

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»

Теплоенергетичний факультет

Кафедра автоматизації проектування енергетичних процесів і систем

До захисту допущено

В.о. завідувача кафедри

\_\_\_\_\_  
(підпис) О.В. Коваль  
(ініціали, прізвище)

“ \_\_\_\_ ” \_\_\_\_\_ 2019р.

**ДИПЛОМНА РОБОТА**  
**на здобуття ступеня бакалавра**

з напрямку підготовки 122 «Комп’ютерні науки та інформаційні технології»  
на тему Реалізація програмного рішення інформаційної системи з  
документообігу в страховій компанії

Виконав: студент 4 курсу, групи ТМ-61

\_\_\_\_\_  
Сіколенко Едуард Володимирович  
(прізвище, ім’я, по батькові) \_\_\_\_\_  
(підпис)

Керівник к.е.н., доцент Сегеда І.В.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали) \_\_\_\_\_  
(підпис)

Рецензент к.е.н., доцент Бريدун Є.В.  
(посада, вчене звання, науковий ступінь, прізвище та ініціали) \_\_\_\_\_  
(підпис)

Засвідчую, що у цій дипломній роботі немає  
запозичень з праць інших авторів без  
відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2020

**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет теплоенергетичний

Кафедра автоматизації проектування енергетичних процесів і систем

Рівень вищої освіти перший, бакалаврський

Напрямок підготовки 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

\_\_\_\_\_ О.В. Коваль  
(підпис)

” \_\_\_\_ ” \_\_\_\_\_ 2020р.

**ЗАВДАННЯ**

**на дипломну роботу студенту**

Сіколенко Едуарду Володимировичу

(прізвище, ім'я, по батькові)

1. Тема роботи Реалізація програмного рішення інформаційної системи з документообігу в страховій компанії

керівник роботи

к.е.н., доцент Сегеда Ірина Василівна

(прізвище, ім'я, по батькові науковий ступінь, вчене звання)

затверджена наказом вищого навчального закладу від « \_\_\_\_ » \_\_\_\_\_ р. № \_\_\_\_\_

2. Строк подання студентом роботи « \_\_\_\_ » \_\_\_\_\_ р.

3. Вихідні дані до роботи: мова програмування — PHP, Java Script, середовище розробки — nano.

4. Зміст розрахунково-пояснювальної записки (перелік завдань, які потрібно розробити) провести глибокий аналіз предметної області та знайти процеси, які можна оптимізувати або автоматизувати. Розробити архітектуру системи та алгоритми її роботи, після чого створити готовий програмний продукт. Запустити тестування та промислову експлуатацію. За результатами роботи зробити висновки.

5. Перелік ілюстративного матеріалу

1. Актуальність роботи. 2-3. Мета та задачі. 4. Архітектура системи.

5. Концептуальна схема бази даних. 6. Процес передачі бланків. 7. Засоби розробки.

8-10. Користувачський інтерфейс. 11-12. Приклад звітних документів. 13. Висновки.

6. Дата видачі завдання «1» грудня 2020 р.

## КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів виконання дипломної роботи	Термін виконання етапів роботи	Примітки
1.	Вивчення та аналіз задачі	10.02.2020 – 02.03.2020	
2	Розробка архітектури та загальної структури системи	03.03.2020 – 05.04.2010	
3.	Розробка структур окремих підсистем	06.04.2020 – 20.04.2020	
4.	Програмна реалізація системи	21.04.2020 – 10.05.2020	
5.	Оформлення пояснювальної записки	11.05.2020 – 31.05.2020	
6.	Захист програмного продукту	08.06.2020	
7.	Передзахист	08.06.2020	
8.	Захист	15.06.2020	

Студент

\_\_\_\_\_  
(підпис)

Сіколенко Е.В.  
(прізвище та ініціали,)

Керівник роботи

\_\_\_\_\_  
(підпис)

Сегеда І.В.  
(прізвище та ініціали,)

## **АНОТАЦІЯ**

Метою цієї роботи є створення системи контролю документообігу в страховій компанії

Робота складається з вступу, чотирьох розділів, висновку, списку використаної літератури та додатків.

Загальний обсяг роботи: 53 сторінки, 35 ілюстрацій, 2 таблиці та 12 бібліографічних найменувань.

Ключові слова: документообіг, система контролю, страхова компанія, база даних, програмне рішення

## **ABSTRACT**

The purpose of this work is to create document control system in the insurance company.

This work consist of introduction, four sections, conclusion and reference list.

Total volume of work: 53 pages, 35 illustrations, 2 tables and 12 references.

Key words: document workflow, control system, insurance company , database, software solution.

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ.....	7
ВСТУП.....	9
1 ПОСТАНОВКА ЗАДАЧІ РЕАЛІЗАЦІЇ ПРОГРАМНОГО РІШЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ З ДОКУМЕНТООБІГУ В СТРАХОВІЙ КОМПАНІЇ.....	11
1.1 Мета створення інформаційної системи з документообігу в страховій компанії.....	11
1.2 Завдання, які потрібно розв’язати для досягнення мети.....	11
1.3 Вимоги до розробленої системи .....	13
2 АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ПРОДУКТІВ .....	15
2.1 Передумови до створення систем з документообігу .....	14
2.1 Порівняння існуючих рішень.....	14
3 ЗАСОБИ РОЗРОБКИ .....	17
3.1 Архітектура програмного забезпечення .....	17
3.2 Вибір серверу .....	17
3.3 Вибір програмних додатків .....	18
3.4 Вибір мов програмування.....	21
3.5 Опис архітектури мережі.....	24
4 ОПИС РЕАЛІЗАЦІЇ.....	27
4.1 Налаштування мережі .....	27
4.2 Аутентифікація та авторизація в системі.....	29
4.3 Створення бази даних .....	31

4.4 Реалізація програмного продукту .....	36
4.4.1 Реалізація створення бланків .....	36
4.4.2 Реалізація передачі бланків .....	36
4.4.3 Реалізація створення акту передачі бланків .....	37
4.4.4 Реалізація перегляду історії по бланку .....	39
4.4.5 Реалізація створення звіту за фільтрами .....	39
4.4.6 Реалізація створення та перегляду реєстру зіпсованих бланків .....	41
4.4.7 Реалізація створення договору страхування .....	44
4.4.8 Реалізація пошуку договору страхування .....	48
4.4.9 Реалізація друку договору страхування та пов'язаних документів .....	49
4.4.10 Реалізація створення звіту по договорах страхування .....	51
 5 ВИСНОВОК .....	53
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ .....	54
ДОДАТОК А .....	54
ДОДАТОК Б .....	58
ДОДАТОК В .....	71

## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СКОРОЧЕНЬ ТА ТЕРМІНІВ

ІТ	— інформаційні технології.
СК	— страхова компанія.
ПІБ	— прізвище ім'я та по-батькові.
ПІН	— індивідуальний податковий номер.
МТСБУ	— моторне (транспортне) страхове бюро України.
Бланк, БСЗ	— бланк суворої звітності.
Договір	— договір страхування.
Web	— Всесвітня мережа.
API	— Application Programming Interface – опис можливостей завдяки яким одна програма може взаємодіяти з іншою.
AWS	— Amazon Web Services, платформа для хмарних обчислень.
EC2	— Elastic Compute Cloud, сервіс AWS який надає хмарні обчислювальні потужності
S3	— Simple Storage Service, сервіс-сховище даних для AWS.
VPN	— Virtual Private Network, віртуальна приватна мережа.
P12	— один з стандартів для зберігання декількох криптографічних об'єктів в одному файлі.
HMAC	— Hash-based message authentication code – механізм перевірки цілісності даних, що передаються.
SSH	— Secure Shell - мережевий протокол рівня застосунків для віддаленого керування комп'ютером
TLS	— Transport Layer Security, протокол який забезпечує захищену передачу даних.
БД	— база даних.

СКБД	— система керування базами даних.
CSV	— Comma-Separated Values, текстовий формат збереження табличних даних.
PDF	— формат файлу для представлення документів у незалежному від пристрою виведення вигляді.
DOC	— формат файлу для збереження текстових даних.
XML	— стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками.



## ВСТУП

Страхові компанії сьогодні користуються великим попитом перш за все через те, що люди не хочуть нести збитки. Портфелі страховиків налічують до 120 страхових продуктів[2]. Відповідно до звіту нацкомфінпослуг [18] в Україні щодня виписується приблизно 167 000 договорів. Кожен з них потрібно укласти, перевірити, відмітити оплату договору, створити акти виконаних робіт, поставити печатки, тощо.

З початку XXI століття люди намагаються полегшити своє життя за допомогою комп'ютерних технологій. В Україні стрімко розвивається електронний документообіг, адже це перш за все швидкість передачі інформації та економія природних ресурсів. Зараз вже не потрібно бути присутнім при укладанні договорів: достатньо мати електронний цифровий підпис, який є цифровим аналогом вашого підпису.

В більшості СК намагаються автоматизувати та оптимізувати більшість бізнес процесів за допомогою різноманітних інформаційних систем, адже їм дуже важливо швидко зберігати та знаходити інформацію. Саме тому страхові компанії зараз звертаються до ІТ-компаній для створення програмних продуктів, які будуть вести контроль та облік за бланками та укладеними договорами, автоматизовувати певні процеси, такі як створення актів, друк договору, відображення статистики по компанії, співробітникам, обчислення фінансових показників СК та формування різноманітних звітів [9].

Станом на перший квартал 2020 року в Україні налічується 225 страхових компаній, однак реальних реалізованих програмних рішень для управління ними та обліку документів можна полічити на пальцях однієї руки. Страхові компанії тільки починають використовувати ІТ в своїй діяльності, тому ця робота має високий ступінь актуальності.

До створення цієї системи страховики вручну укладали всі договори страхування, а їх облік вели в програмі Microsoft Excel. Файлом-реєстром неможливо користуватись декільком людям одночасно, адже в такому випадку існує загроза втрати даних. До того ж через велику кількість договорів програма «зависала», довго завантажувалась та займала багато обсягу на диску.

Система для контролю документообігу в страховій компанії здатна вирішити всі ці проблеми, адже нею можуть одночасно користувались велика кількість людей, користувачу відображається лише необхідна інформація, а за допомогою правильного налаштування бази даних та індексації полів швидкість пошуку інформації значно зросла.

Враховуючи те, що страховики вважають свою предметну область найскладнішою для розробників програмного забезпечення через те, що вона включає в себе велику кількість різноманітних процесів та значні обсяги даних з різних галузей – більшість розроблених систем є модульними, та можуть використовуватись в інших предметних областях.

# **1 ПОСТАНОВКА ЗАДАЧІ РЕАЛІЗАЦІЇ ПРОГРАМНОГО РІШЕННЯ ІНФОРМАЦІЙНОЇ СИСТЕМИ З ДОКУМЕНТООБІГУ В СТРАХОВІЙ КОМПАНІЇ**

## **1.1 Мета створення інформаційної системи з документообігу в страховій компанії**

В ХХІ столітті людство почало використовувати електронно обчислювальні машини не тільки в наукових цілях, але й в керуванні державою, бізнесом та в своїх особистих цілях. Саме тому в світі почав активно використовуватись термін «Діджиталізація». Вже сьогодні кількість відправленої електронної пошти значно перевищує кількість відправлень традиційною поштою.

Враховуючи велику кількість інформації, яку обробляють люди в СК, питання використання сучасних технологій постає досить гостро. Саме тому мета цієї дипломної роботи є оптимізація та цифровізація документообігу в страховій компанії.

Проблема полягає в тому, що в теперішній час системи, які надають можливість контролювати документообіг в страховій компанії занадто дорогі, та при цьому не виконують весь потрібний функціонал. Страховики почали використовувати програму Microsoft Excel для контролю документообігу, однак постало питання одночасного користування файлом декількома співробітниками та розмежування прав доступу.

Враховуючи вищезазначені фактори була поставлена задача створити програмну реалізацію інформаційної системи з документообігу в страховій компанії. Ця система повинна розподілити роботу користувачів в системі відповідно до їх службових обов'язків, надати можливість одночасного користування та редагування інформації, автоматично створювати звіти та заповнювати договори страхування з можливістю їх друку.

Потенційними користувачами є головний офіс, регіональні підрозділи та агенти страхових компаній.

## **1.2 Завдання, які потрібно розв'язати для досягнення мети**

Для виконання цієї роботи потрібно було досить детально розглянути предметну область та зрозуміти які саме процеси можна оптимізувати та автоматизувати. Для цього важливо розуміти службові обов'язки людей в компанії та життєвий цикл договору. Відповідно до цього було сформовано технічне завдання та поставлені наступні задачі до програмного продукту:

- спроектувати систему з додержанням основних вимог інформаційної безпеки;
- забезпечити можливість користування системою з будь-якої точки світу;
- розробити систему авторизації та аутентифікації користувачів. Чітко визначити ролі та можливості користувачів у системі;
- проаналізувати предметну область та створити потрібні довідники;
- розробити та реалізувати алгоритм контролю за бланками суворої звітності з збереженням всіх змін та інформації хто та коли їх зробив;
- забезпечити можливість створювати акти передачі бланків та автоматично присвоювати їм внутрішній номер при зміні користувача бланку з можливістю додавати декілька різних серій та діапазонів номерів;
- надати можливість створювати договори страхування на основі бланку. При

- виборі бланку автоматично знаходити в базі даних інформацію стосовно компанії та користувача. Розробити алгоритм пошуку клієнта в базі даних за ПІН. Реалізувати калькулятор для договорів ОСЦПВ за вказаними даними;
- розробити можливість пошуку договору за номером, ПІН або ПІБ страхувальника;
  - створити пошук договорів по різноманітним фільтрам із можливістю виведення інформації на екран та у CSV;
  - реалізувати можливість генерування PDF файлів для друку договору страхування на бланку;
  - автоматизувати процес створення акту виконаних робіт, рахунку фактури та акту передачі бланків шляхом автоматичного їх створення в форматі DOC;
  - при створенні програмного продукту дотримуватись кольорової гами компанії, використовувати затверджені брендбуком шрифти.

### **1.3 Вимоги до розробленої системи**

Під час формування технічного завдання однією з умов було забезпечити можливість користуватись системою з будь-якої точки світу в будь-який час та можливість одночасного користування системою різними клієнтами.

Основна проблема полягала в створенні безпечного доступу до системи та унеможливити заволодіння даними третіми особами. Для розв'язання цієї проблеми було вирішено використовувати VPN сервер. Для роботи з системою клієнт повинен встановити додаток для встановлення VPN з'єднання та підключитись до мережі за допомогою ключа-сертифіката P12, та файлу перевірки цілісності інформації НМАС.

Також важливо було надати доступ до системи з різноманітних пристроїв, в тому числі мобільних телефонів або планшетів, тому для уніфікації версій було прийняте рішення розроблювати програмний продукт з web-інтерфейсом. Отже для використання потрібно встановити будь-який вебпереглядач.

## **2 АНАЛІЗ ІСНУЮЧИХ ПРОГРАМНИХ ПРОДУКТІВ**

### **2.1 Передумови до створення систем з документообігу**

В Україні стрімко зменшується кількість страхових компаній. Лише за 2019 рік було закрито 49 СК [1]. Це пов'язано з складною економічною ситуацією та політикою національної комісії що здійснює державне регулювання у сфері ринків фінансових послуг. Саме через ці фактори не кожна компанія має можливість розробити власне програмне забезпечення, тому вони вимушені вести документообіг в програмах на кшталт «Microsoft Excel» та розробляти власні форми актів, змінюючи їх вручну, або платити гроші за налаштування програми 1С для страхової компанії, яка до того ж з 2017 року заборонена для поширення в Україні [19].

### **2.2 Порівняння існуючих рішень**

Однак на ринку існують і спеціально створені програми для страховиків. До таких програм можна віднести EWA та ProfITsoft.

Перша програма досить добре зарекомендувала себе в якості помічника страховику: її використовують 45 страхових компаній, однак вона досить навантажена та складна в користуванні. Для створення різноманітних актів або створення файлів в форматі PDF потрібно мати навички в програмуванні та встановлювати додаткове програмне забезпечення. Також не завжди вийде налаштувати страховий продукт так, щоб він працював так як потрібно, адже в системі не передбачені важливі атрибути договору, наприклад франшиза. Проте розробники надали можливість додавати власні атрибути, але при створенні звіту по цим договорам дані з цих полів не відображаються. Основна перевага цього сервісу – можливість налаштування зв'язку з сайтом компанії завдяки API

ProfITsoft в свою чергу досить добре оптимізована, однак це рішення лише для бекофісу, тому її неможливо підключити до сайту компанії для продажу електронних договорів страхування.

Також слід розуміти що використовуючи подібні сервіси існує ризик витоку інформації третім особам, адже це програмне забезпечення з закритим сирцевим кодом, та невідомо хто має доступ до баз даних.

Слід згадати і фінансову частину питання. На даний момент вартість оренди програмного продукту ProfITsoft – 29 000 грн/місяць, але орендувати можна мінімум на три роки, отже витрати за перший місяць будуть більше одного мільйона гривень. Його також можна купити повністю. Вартість всього програмного забезпечення – 1 300 000 грн, але оновлення системи будуть доступні лише перші два роки.

EWA в свою чергу пропонує більш привабливі умови: 10 000 грн/місяць та відсоток від страхової премії, однак варто розуміти що у разі, коли у компанії великий документообіг, вартість оренди буде значно рости.



## **3 ЗАСОБИ РОЗРОБКИ**

### **3.1 Архітектура програмного забезпечення**

Додаток створений як веб-застосунок – розподілений застосунок, в якому клієнтом виступає браузер, а сервером – веб-сервер. Так як клієнти можуть підключатись до системи з мобільних пристроїв та з своїх домашніх комп'ютерів або ноутбуків, потужність яких може бути незадовільною для виконання складних обчислень та обробки великої кількості даних, а в страхуванні по іншому не виходить, було прийняте рішення розробки «тонкого клієнта». Це означає що всі обчислення зосереджуються на веб-сервері.

### **3.2 Вибір серверу**

В якості веб серверу було обрано «Amazon Web Services» з їх віртуальною машиною EC2.

Обчислювальний хмарний сервіс Amazon EC2 – це веб-сервіс, що надає безпечні масштабовані обчислювальні ресурси в хмарі. Він допомагає розробникам, полегшуючи проведення обчислень в хмарі в масштабі всього Інтернету. Простий веб-інтерфейс сервісу Amazon EC2 дозволяє отримати доступ до обчислювальних ресурсів і налаштувати їх з мінімальними зусиллями. Він надає користувачам повний контроль над обчислювальними ресурсами, а також перевірену обчислювальну середу Amazon для роботи .

Використовуючи AWS питання безпеки також делегуються до Amazon. Крім того, при реєстрації Amazon надає безкоштовне використання одного серверу протягом одного року.

На серверах EC2 встановлена операційна система Linux, що дуже схожа на образ CentOS7.

Варто розуміти що підключення до серверу відбувається завдяки протоколу SSH, тому всі зміни та налаштування відбуваються за допомогою терміналу та програмного забезпечення PuTTY. Інтерфейс програми PuTTY зображений на рисунку 3.1.

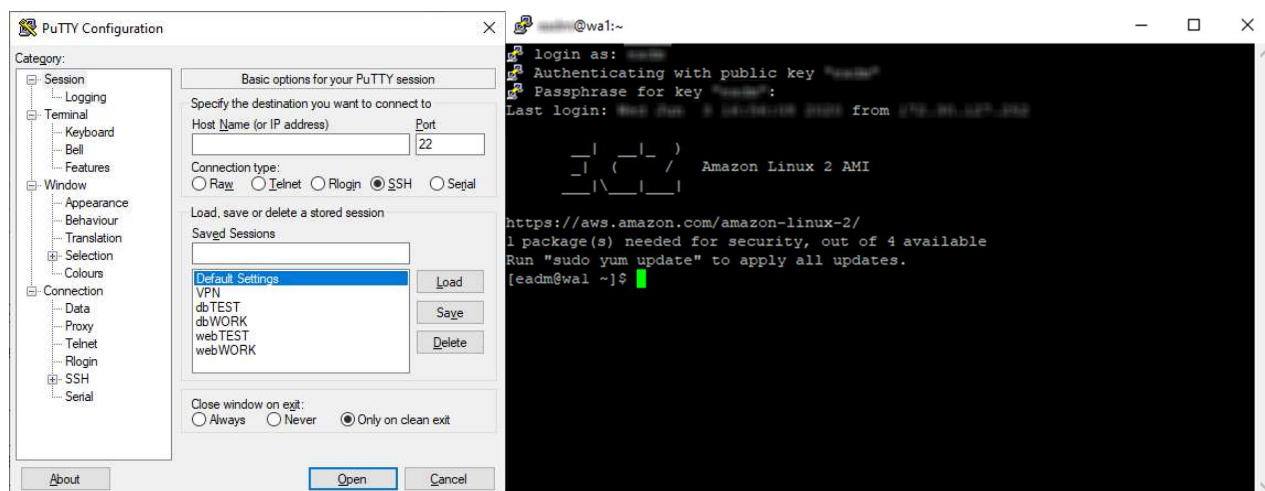


Рисунок 3.1 – Інтерфейс програми PuTTY

### 3.3 Вибір програмних додатків

Сьогодні більшість програмних продуктів є платними, однак існує достатньо програм з відкритим сирцевим кодом. Це дає можливість отримувати нові знання, поліпшувати вже створений продукт, співпрацювати з розробниками. Безпека – також важливий фактор, адже можна переглянути код та бути впевненим що всі дані користувача будуть в безпеці. Проте найголовніше в програмах з відкритим сирцевим кодом – це те, що не потрібно витратити кошти на ліцензії.

VPN мережа створена для підвищення рівня безпеки, адже всі клієнти розмежовані і не мають доступ до комп'ютерів іншої мережі. Однак це і можливість об'єднати їх в одній мережі. Загальна схема використання зображена на рисунку 3.2.

Для створення VPN мережі використовується OpenVPN Server. OpenVPN – вільна реалізація технології віртуальної приватної мережі (VPN) з відкритим

початковим кодом для створення шифрованих з'єднань між двома клієнтськими машинами або забезпечення роботи централізованого VPN-сервера для одночасної роботи декількох клієнтів. OpenVPN дозволяє встановлювати з'єднання між комп'ютерами, що перебувають за NAT-екраном, без необхідності зміни їхніх налаштувань. Для забезпечення безпеки керуючого каналу і потоку даних OpenVPN використовує бібліотеку OpenSSL. Завдяки цьому залучено весь набір алгоритмів шифрування, доступних у цій бібліотеці. Також може використовуватися пакетна авторизація HMAC для забезпечення вищого рівня безпеки.

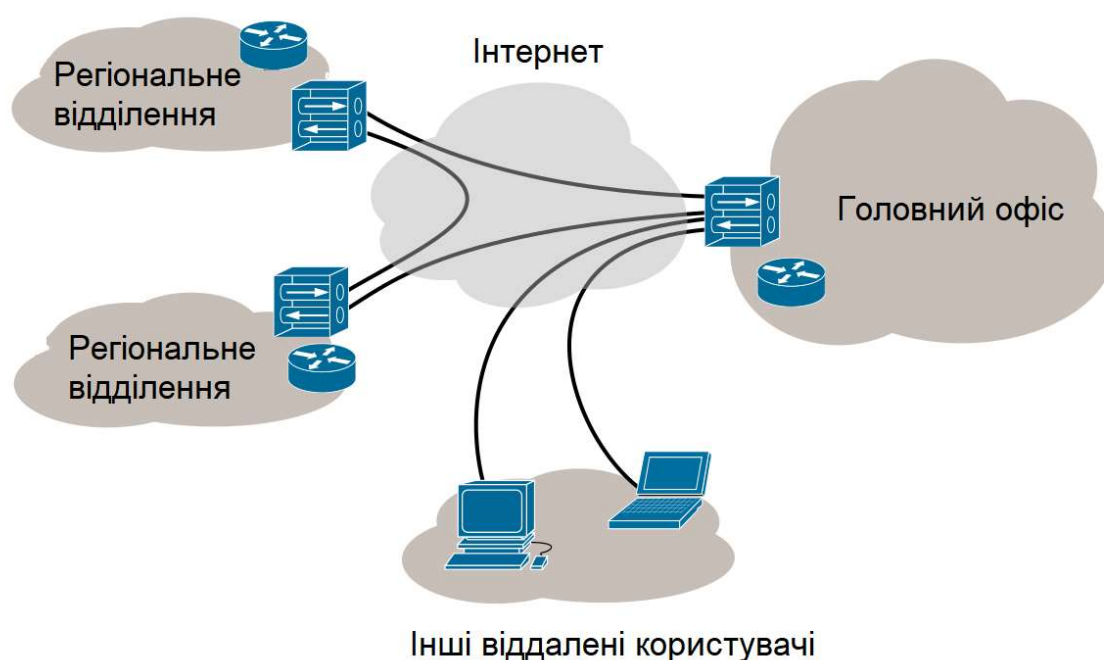


Рисунок 3.2 – Загальна схема використання VPN

В якості веб серверу був обраний Apache 2.4.41. Apache HTTP-сервер – відкритий веб-сервер Інтернет для UNIX-подібних, Microsoft Windows, Novell NetWare та інших операційних систем. Apache розроблюється та підтримується спільнотою розробників відкритого програмного забезпечення під керівництвом Apache Software Foundation. В 1996 році Apache обійшов NCSA HTTPd із того часу є найбільш популярним веб-сервером у світі. Станом на червень 2013 року Apache був

встановлений на 53,34% серверів (358 974 045 одиниць). Для порівняння: на другому місці Microsoft IIS - їхня частка становить 17,22% серверів (115 920 681 одиниць). Web-сервер Apache є самостійним, некомерційним, вільно розповсюджуваним продуктом. Продукт підтримує безліч можливостей, багато з яких реалізовані як скомпільовані модулі, які розширюють основні функціональні можливості. Існують інтерфейси для підтримки мов програмування Perl, Python, Tcl і PHP. Функції віртуального хостингу дозволяють одній інсталяції Apache обслуговувати різні веб-сайти. На цей веб сервер досить легко поставити SSL сертифікат безпеки для того, щоб клієнт обмінювався даними з сервером по захищеному 443 порту.

В якості СУБД було обрано MariaDB 5.5.64. MariaDB – реляційна система керування базами даних, створена на початку 2009 року як відгалуження MySQL. MariaDB поширюється під вільною та відкритою ліцензією GNU GPL. Поштовхом для створення MariaDB стала непевність спільноти розробників та користувачів невизначеною долею і ліцензією MySQL після її придбання Oracle. Проект підтримує високий ступінь сумісності з MySQL , включаючи "вставку" замінних можливостей бінарним еквівалентом бібліотек і відповідністю з наявним API і командами MySQL. Навіть перша стабільна версія СУБД отримала номер 5.2, ніби продовжуючи лінійку MySQL.

Для полегшення роботи було встановлено програмне забезпечення phpMyAdmin. Воно дає можливість швидко та легко змінювати дані і БД не використовуючи термінал та SQL команди. Інтерфейс програми зображений на рисунку 3.3.

Для генерації PDF-документів використовується безкоштовна бібліотека з відкритим кодом TCPDF. TCPDF – це єдина бібліотека на основі PHP, яка включає повну підтримку Unicode UTF-8. Зазначена бібліотека є однією з найбільш використовуваних бібліотек PHP у світі, завдяки тому, що вона вже включена до найпопулярніших CMS додатків.

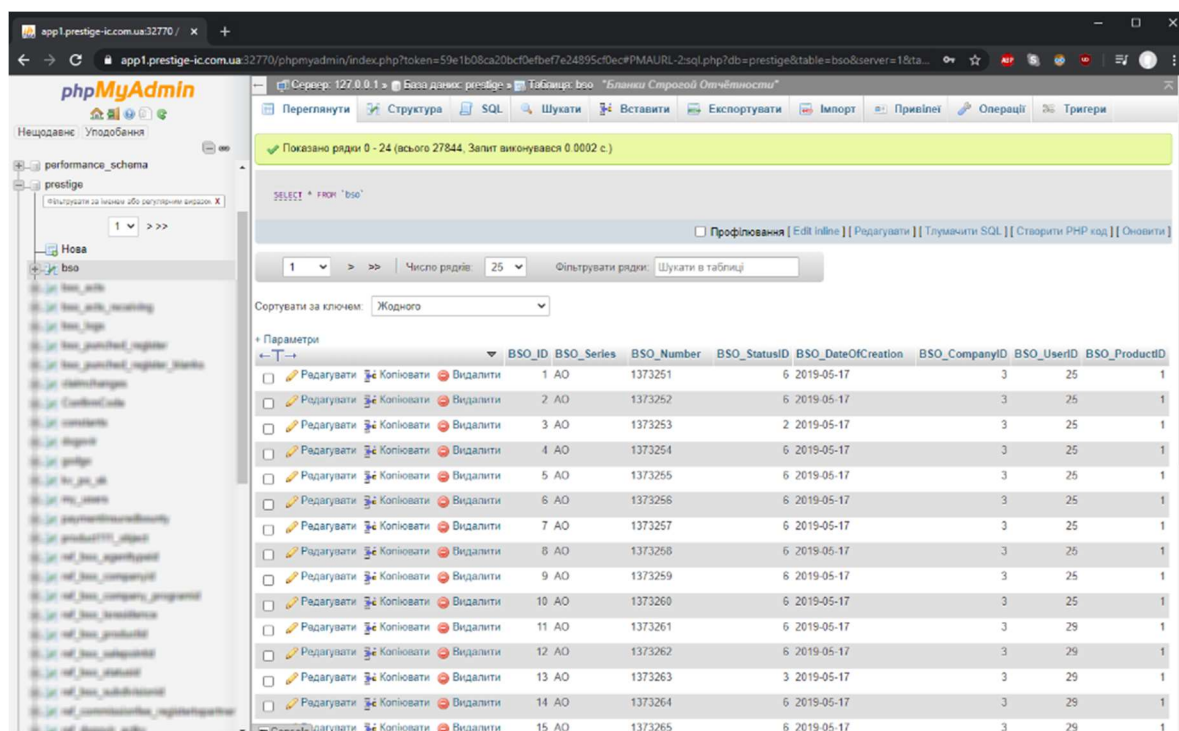


Рисунок 3.3 – Інтерфейс програми phpMyAdmin

### 3.4 Вибір мов програмування

В якості мови програмування було обрано JavaScript з бібліотекою jQuery та PHP.

Таке рішення обумовлене тим, що JavaScript є єдиною динамічною, об'єктно-орієнтованою мовою програмування, скрипти якої виконуються на стороні клієнта (пристрої кінцевого користувача). Вона найчастіше використовується для створення сценаріїв веб-сторінок, що надає можливість на стороні клієнта взаємодіяти з користувачем, керувати браузером, асинхронно обмінюватися даними з сервером, змінювати структуру та зовнішній вигляд веб-сторінки. JavaScript класифікують як прототипну, скриптову мову програмування з динамічною типізацією. Особливістю JavaScript є те, що вона має C-подібний синтаксис.

jQuery – популярна JavaScript-бібліотека з відкритим кодом. Вона була представлена у січні 2006 року. Згідно з дослідженнями організації W3Techs, JQuery

використовується понад половиною найвідвідуваніших сайтів. JQuery є найпопулярнішою бібліотекою JavaScript, яка посилено використовується на сьогоднішній день. JQuery є вільним програмним забезпеченням під ліцензією MIT. Синтаксис jQuery розроблений, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, створенню анімації, обробки подій, і розробки AJAX-застосунків. Основне завдання jQuery – це надавати розробнику легкий та гнучкий інструментарій кросбраузерної адресації DOM об'єктів за допомогою CSS та XPath селекторів.

Для написання також використовувалась мова розмітки HTML. Це мова тегів, якою пишуться гіпертекстові документи для мережі Інтернет. Веб-браузери отримують HTML-документи з веб-сервера або з локальної пам'яті і передають документи в мультимедійні веб-сторінки.

В свою чергу на стороні серверу використана скритована мова PHP. Вона є однією з найпоширеніших мов, що використовуються у сфері веб-розробок. PHP підтримується переважною більшістю хостинг-провайдерів. PHP – проект відкритого програмного забезпечення. PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, тому що браузер отримує готовий HTML-код. Це є перевагою з точки зору безпеки, але погіршує інтерактивність сторінок. Але надана можливість використовувати мову PHP для генерування JavaScript-кодів, які виконуються вже на стороні клієнта.

Для оформлення сторінок використовувалась мова CSS – спеціальна мова стилю сторінок, що використовується для опису їхнього зовнішнього вигляду. CSS є основною технологією всесвітньої павутини, поряд із HTML та JavaScript. Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

Всі дані передаються або формами HTML, або JSON чи XML файлами.

За допомогою форм HTML можна передавати дані POST та GET запитами.

POST — один з багатьох методів запиту, що підтримуються протоколом передачі даних HTTP, який використовується у всесвітній мережі Інтернет. Метод POST призначений для запиту, при якому веб-сервер приймає дані збережені в тілі повідомлення, для зберігання. Метод часто використовується для завантаження файлу або передачі заповненої веб-форми. На відміну від POST, метод GET призначений для отримання інформації від сервера. В рамках GET-запитів деяких даних можуть бути передані в рядку запиту URI, який вказує, наприклад, умови пошуку, діапазони дати, або іншу інформацію, що визначає запит. В рамках методу POST-запиту довільну кількість даних будь-якого типу може бути відправлено на сервер в тексті листа запиту.

JSON — це текстовий формат обміну даними між комп'ютерами. JSON базується на тексті, може бути прочитаним людиною. Формат дає змогу описувати об'єкти та інші структури даних. Цей формат використовується переважно для передачі структурованої інформації через мережу.

XML — запропонований консорціумом World Wide Web Consortium стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема, через Інтернет. XML-документ складається із текстових знаків, і придатний до читання людиною.

JSON знайшов своє головне призначення в написанні веб-програм, а саме при використанні технології AJAX. JSON, що використовується в AJAX, виступає як заміна XML під час асинхронної передачі структурованої інформації між клієнтом та сервером. При цьому перевагою JSON перед XML є те, що він дозволяє використовувати складні структури в атрибутах, займає менше місця і прямо інтерпретується за допомогою JavaScript в об'єкти.

AJAX — підхід до побудови користувацьких інтерфейсів веб-застосунків, за яких вебсторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звідти довантажує потрібні користувачу дані. AJAX — один з компонентів концепції DHTML. Процес роботи AJAX зображений на рисунку 3.4.

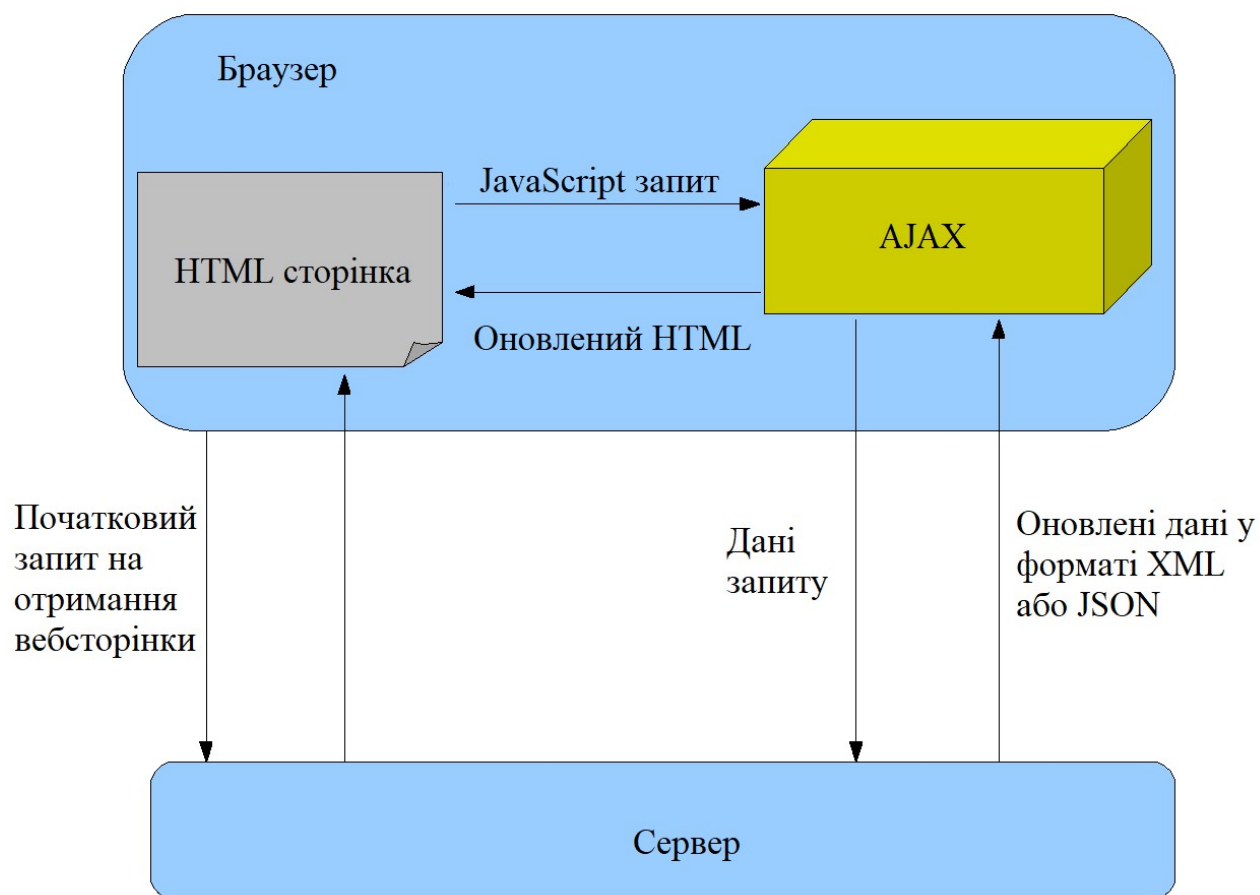


Рисунок 3.4 – Схема роботи AJAX

Порівняємо дві моделі: класичну та з використанням AJAX

Класична модель веб-застосунку:

- Користувач заходить на вебсторінку і натискає на який-небудь її елемент
- Браузер надсилає запит серверу
- У відповідь сервер генерує повністю нову вебсторінку і відправляє її браузеру і т. д.

Модель AJAX:

- Користувач заходить на вебсторінку і натискає на який-небудь її елемент.
- Браузер відправляє відповідний запит на сервер.
- Сервер віддає тільки ту частину документа, яка змінилася.



### 3.5 Опис архітектури мережі

Загальна архітектура мережі зображена на рисунку 3.5. IP адреси не відображаються через вимоги безпеки.

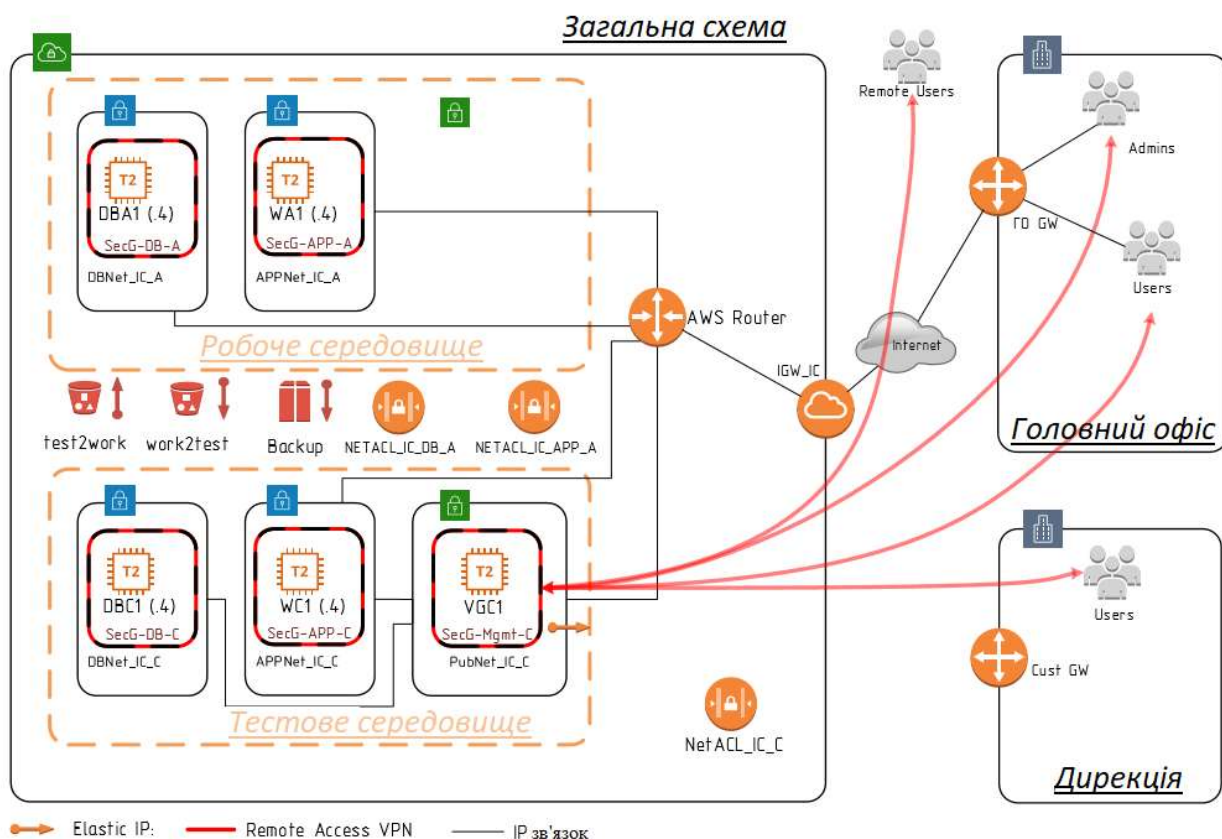


Рисунок 3.5 – Загальна схема інфраструктури

На схемі присутні два середовища – тестове та робоче. Це зроблено для того, щоб клієнти працювали на стабільній версії програми. Обмін файлами між середовищами відбувається за допомогою Amazon S3 – сервіс-сховище даних, один з сервісів AWS. Сервіс надає можливість для зберігання й отримання будь-якого обсягу даних, у будь-який час з будь-якої точки мережі, тобто так званий файловий хостинг. За допомогою Amazon S3 досягається висока масштабованість, надійність, висока швидкість, недорога інфраструктура зберігання даних. При виконанні скрипта «work2test» копіюється база даних та програмний код, а при виконанні «test2work» - лише

програмний код, для унеможливлення втрати даних, що були внесені в БД.

В кожному середовищі розміщені дві віртуальні машини Amazon EC2. Це web-сервер (WA1, WC1) та сервер бази даних (DBA1, DBC1). В тестовому середовищі також розміщений VPN сервер. На віртуальній машині WA1 налаштований планувальник задач cron для створення резервних копій програми та бази даних кожні дві години. Ці копії зберігаються протягом двох днів, після чого видаляються. Це дає можливість швидко відновити базу даних у випадку перебоїв на сервері.

На кожному web-сервері встановлено сертифікати безпеки.

## 4 ОПИС РЕАЛІЗАЦІЇ

### 4.1 Налаштування мережі

Для того, щоб почати використовувати Amazon AWS потрібно спочатку зареєструватись на їхньому сайті та додати банківську картку для оплати. Після реєстрації буде надано бонусні 744 години безкоштовної роботи для однієї віртуальної машини на місяць. Рекомендується встановлювати багатофакторну аутентифікацію за допомогою, наприклад, Google Authenticator, адже були випадки коли зловмисники отримували доступ до аккаунту та створювали велику кількість віртуальних машин, після чого реальному власнику приходили космічні рахунки за використання сервісу. [20]

Головна сторінка консолі AWS зображена на рисунку 4.1. Рекомендується винести на верхню панель наступні сервіси:

- VPC – віртуальна хмара в якій знаходяться всі обчислювальні машини. З цього меню можна керувати налаштуваннями безпеки Network ACL та Security Groups.
- EC2 – в цьому меню міститься інформація стосовно доступних віртуальних машин. З нього можна запускати їх та дивитися їх поточний стан та налаштування.
- IAM – менеджер аккаунтів для AWS. Тут створюються нові користувачі для надання доступу до консолі, якщо проект розробляється не однією особою, та встановлюється багатофакторна аутентифікація. Також на цій сторінці можна змінити власний пароль. Amazon рекомендує змінювати пароль раз на 90 днів.
- S3 – Сховище даних для сервісів AWS.

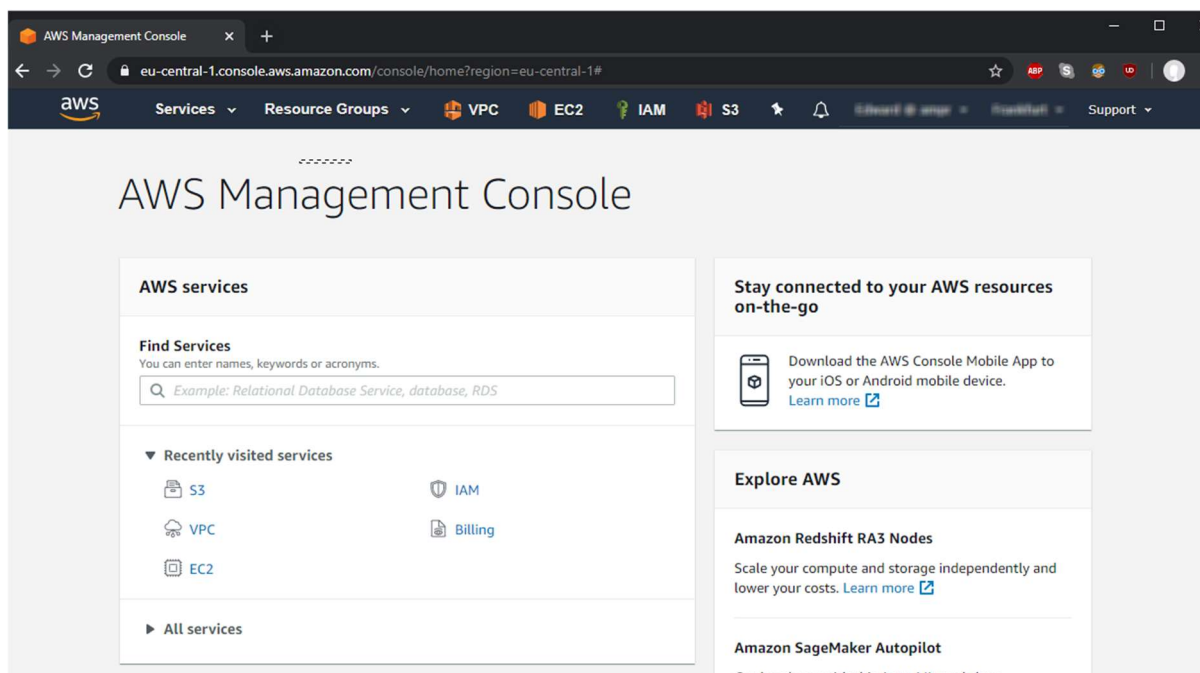


Рисунок 4.1 – Головна сторінка AWS

Як було написано в розділі 3.5 всього створено 5 віртуальних машин, однак лише одна має доступ до мережі Інтернет. На рисунку 4.2 зображено перелік віртуальних машин та частина їх налаштувань. IP адреси та унікальні ідентифікатори не відображаються з вимог безпеки.

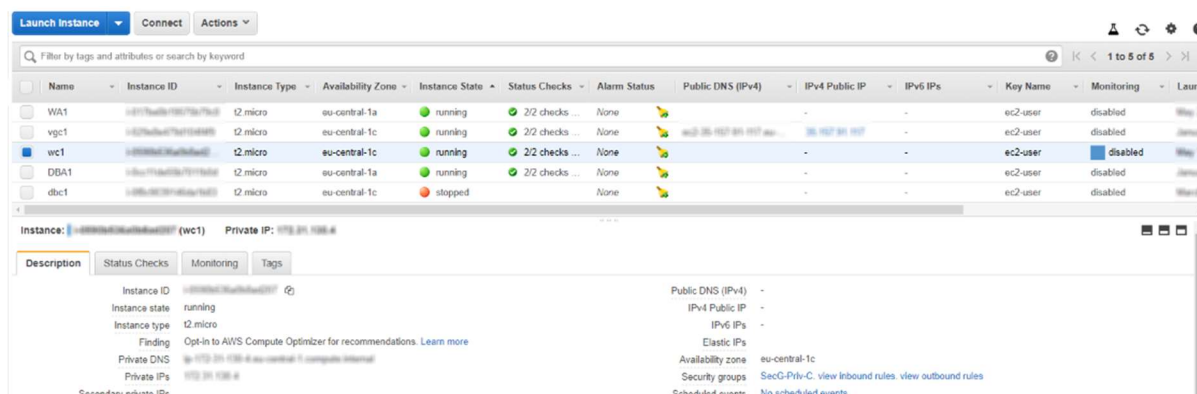


Рисунок 4.2 – Перелік віртуальних машин

В розділі VPC Security Groups було створено декілька груп безпеки. Для кожної віртуальної машини були налаштовані окремі права доступу з вказанням дозволених IP адрес та номери портів, за якими дозволене підключення. Перелік груп зображений на рисунку 4.3, а приклад налаштування - на рисунку 4.4.

Security Groups (7) Info

Filter security groups

Actions

Create security group

<input type="checkbox"/>	Name	Security group ID	Security group name	VPC ID	Description	Owner	Inbound rules count	Outbound
<input type="checkbox"/>	-	sg-01f0a6a6b79a8f5d2	SecG-IC-TEST	vpc-0b7d8a63bada718793c	SecG-IC-TEST	87911898779a23a	1 Permission entry	1 Permissi
<input type="checkbox"/>	SecG-DB-A	sg-0a8778a47e715a-88882	SecG-DB-A	vpc-0b7d8a63bada718793c	SecG-DB-A	87911898779a23a	3 Permission entries	1 Permissi
<input type="checkbox"/>	SecG-APP-C	sg-0a8778a47e715a-29882	SecG-Priv-C	vpc-0b7d8a63bada718793c	launch-wizard-1 cre...	87911898779a23a	10 Permission entries	1 Permissi
<input type="checkbox"/>	-	sg-0b7779afec717a63a23a	default	vpc-0b7d8a63bada718793c	default VPC security ...	87911898779a23a	1 Permission entry	1 Permissi
<input type="checkbox"/>	SecG-DB-C	sg-0b8a63bada718793c-23a	SecG-DB-C	vpc-0b7d8a63bada718793c	launch-wizard-1 cre...	87911898779a23a	4 Permission entries	1 Permissi
<input type="checkbox"/>	SecG-PUB-C	sg-0b8a63bada718793c-7a23a	SecG-Pub-C	vpc-0b7d8a63bada718793c	Security group for P...	87911898779a23a	37 Permission entries	1 Permissi
<input type="checkbox"/>	SecG-APP-A	sg-0ba23779a8bada718793c	SecG-APP-A	vpc-0b7d8a63bada718793c	SecG-APP-A	87911898779a23a	2 Permission entries	1 Permissi

Рисунок 4.3 – Перелік Security Groups

Inbound rules

Edit inbound rules

Type	Protocol	Port range	Source	Description - optional
SSH	TCP	22	172.30.1.27/248/29	Allow RDP/SSH from Admins
SSH	TCP	22	172.31.1.28-4/52	SSH from sg-1
Custom TCP	TCP	21	172.30.1.27/248/29	Allow FTP from Admins
Custom TCP	TCP	52790	172.30.1.27/248/52	-
Custom TCP	TCP	52790	172.30.1.27/252/52	Allow RDP/SSH from Admins
Custom TCP	TCP	20	172.30.1.27/248/29	Allow FTP from Admins
MYSQL/Aurora	TCP	3306	172.31.1.28-4/52	MySQL from sg-1
HTTPS	TCP	443	172.30.1.27-8/24	Allow HTTPS from office
Custom ICMP - IPv4	Echo Request	N/A	172.31.1.28-4/52	Allow ping from sg-1
Custom ICMP - IPv4	Echo Request	N/A	172.30.1.27/248/29	Allow ping from Admins

Рисунок 4.4 – Приклад налаштування Security Groups

Також було створено чотири корзини S3: дві для передачі даних, одна для збереження логів та одна для збереження бекапів системи. Всі корзини приватні, отже доступ до них можна отримати лише увійшовши в аккаунт. Перелік корзин зображений на рисунку 4.5.

S3 buckets

Search for buckets

Create bucket

Edit public access settings

Empty

Delete

<input type="checkbox"/>	Bucket name	Access
<input type="checkbox"/>	prestige-ic.backup	Bucket and objects not public
<input type="checkbox"/>	prestige-ic.logs	Bucket and objects not public
<input type="checkbox"/>	prestige-ic.test2work	Bucket and objects not public
<input type="checkbox"/>	prestige-ic.work2test	Bucket and objects not public

Рисунок 4.5 – Перелік корзин S3

Для мережі був створений корінний сертифікат безпеки, два серверні сертифікати для веб серверів для організації безпечного обміну даних по шифрованим каналам використовуючи 443 порт та 20 сертифікатів P12 для підключення до VPN мережі. Для створення сертифікатів було використано програмне забезпечення OpenSSL.

## **4.2 Аутентифікація та авторизація в системі**

В якості аутентифікації в системі була прийнята аутентифікація в базі даних. Для кожного користувача створено окремий обліковий запис, який має свої власні права доступу. Логін та пароль зберігаються в базі даних mysql, при чому пароль хешований з сіллю, що унеможливлює отримання несанкціонованого доступу до системи в разі витоку хешованого паролю. Якщо користувач підключився до бази даних – він отримує доступ до всієї системи. Після цього користувач отримує свої права, які розписані в базі даних. Вони записуються в змінну сесії PHP. Також користувачу генерується його власне меню, в якому відсутні ті посилання, на використання яких в нього немає прав (наприклад за внесення нових бланків в систему відповідає лише один співробітник, і пункт «Створення» відображається лише йому).

Не дивлячись на розмежування прав в програмному продукті було вирішено розмежувати права доступу і в самій базі даних. Так для деяких користувачів дозволено лише переглядати дані в БД, деякі можуть записувати дані в окремі таблиці. Зміна словників в БД дозволена лише корінному користувачу. Для прикладу, на рисунку 4.6 зображено, що в системі існує користувач для отримання інформації стосовно курсів валют, і він має доступ лише на виконання SELECT та UPDATE запитів на одну таблицю.

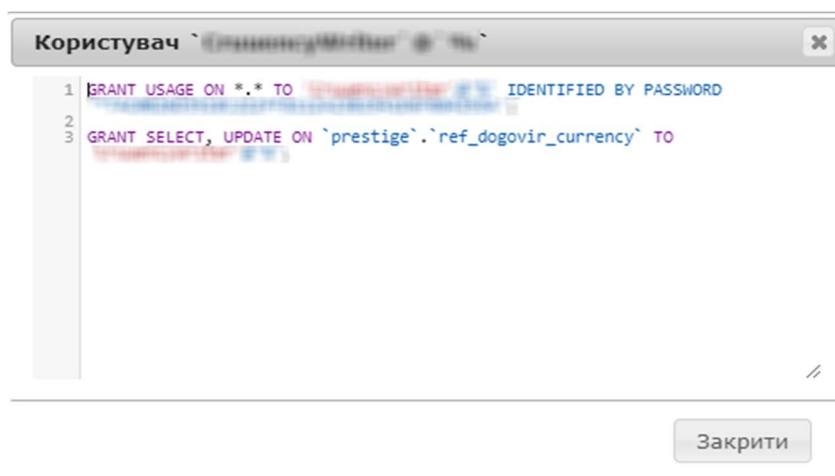


Рисунок 4.6 – Привілеї користувача для запису курсу валют

Після відкриття системи в браузері відкривається сторінка для входу в свій аккаунт. Форма входу зображена на рисунку 4.7, а головна сторінка програми зображена на рисунку 4.8.

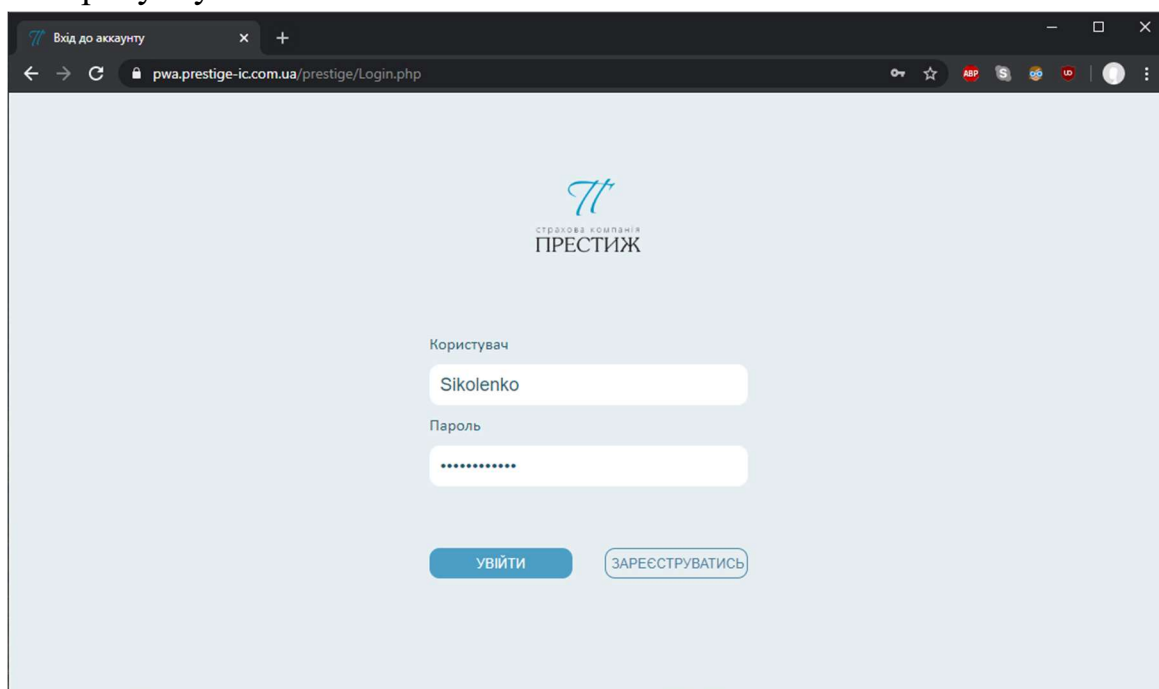


Рисунок 4.7 – Форма входу в систему

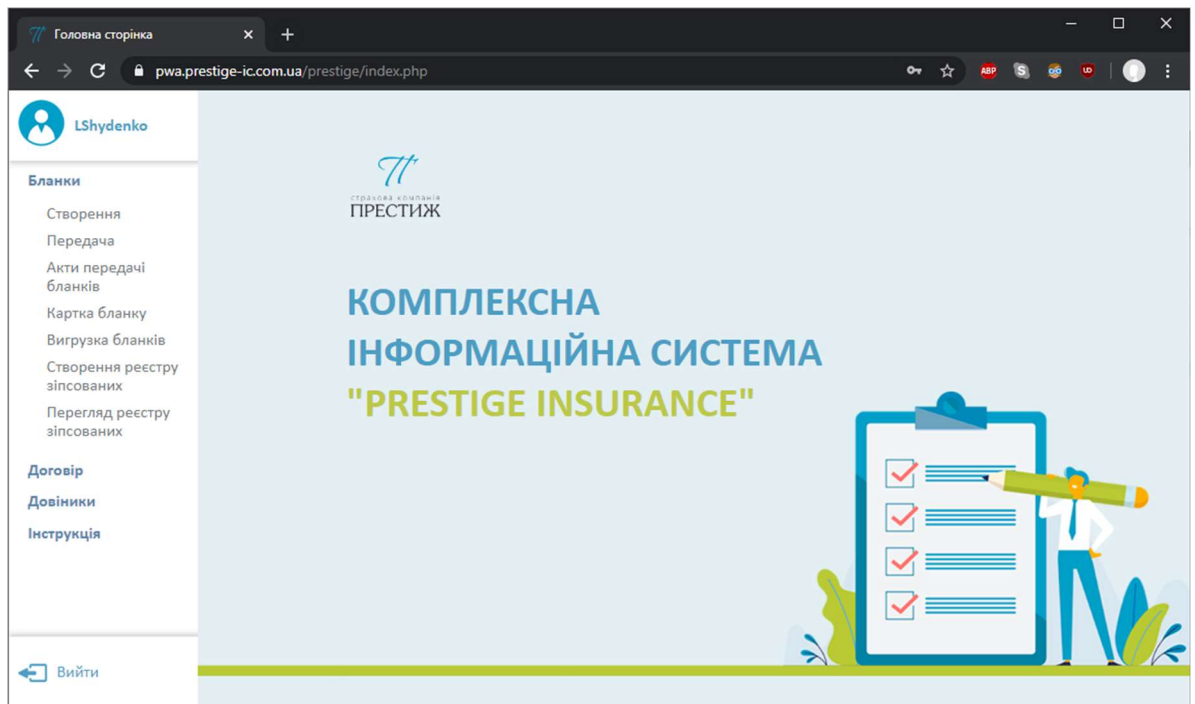


Рисунок 4.8 – Головна сторінка програми

### 4.3 Створення бази даних

В базі даних налічується двадцять дві таблиці. Концептуальна схема бази даних зображена на рисунку 4.9.



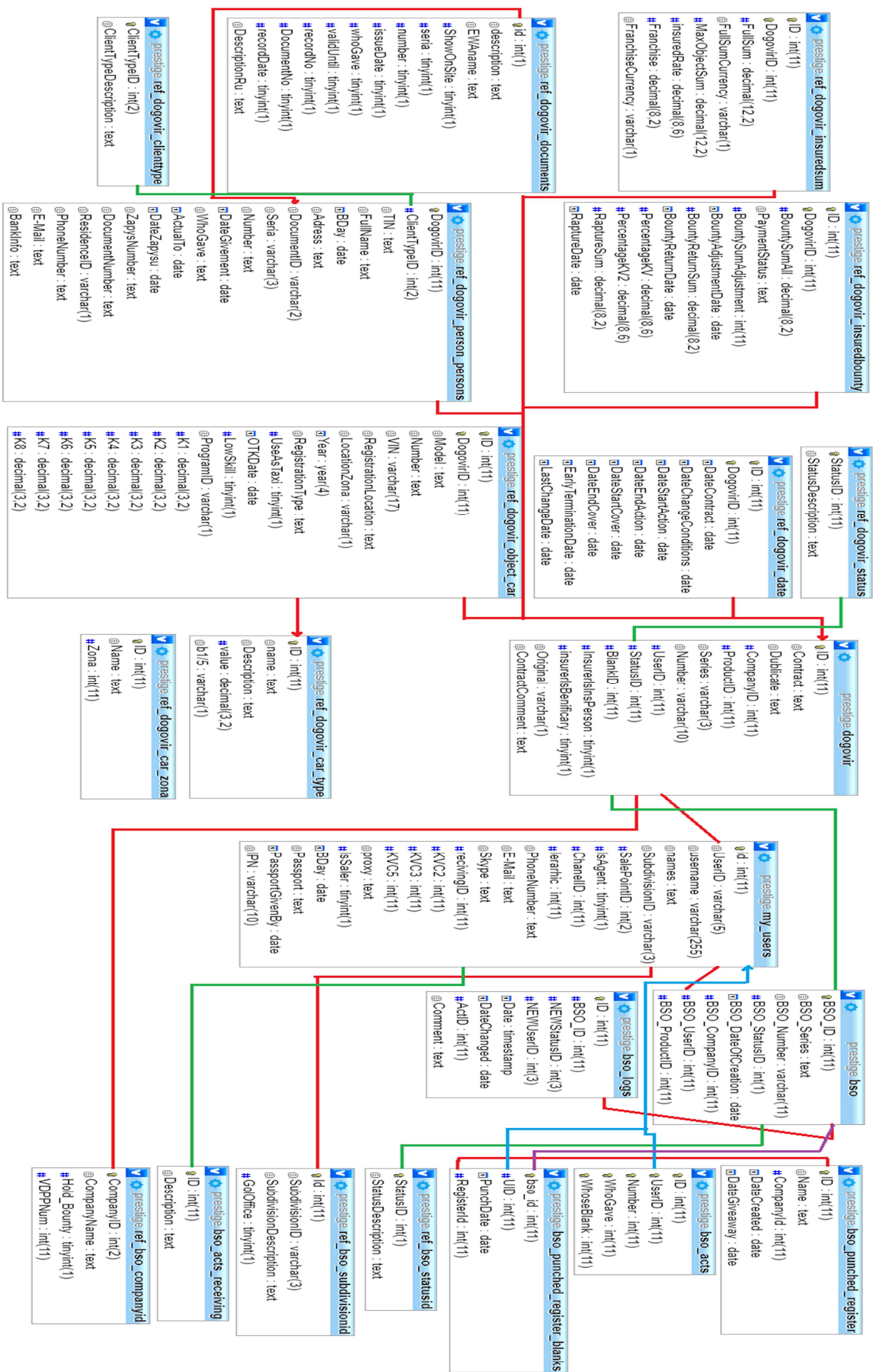


Рисунок 4.9 – Концептуальна схема БД

Опис призначень таблиць в базі даних знаходиться в таблиці 1.

Таблиця 4.1 —Опис призначення таблиць в базі даних

Назва таблиці	Призначення
bso	Інформації про бланк
bso_acts	Акти прийому-передачі бланків
bso_acts_receiving	Посади для актів прийому-передачі
bso_logs	Збереження змін в бланках та прикріплення бланку до акту прийому-передачі
bso_punched_register	Реєстри зіпсованих бланків
bso_punched_register_blanks	Список зіпсованих бланків з датою псування та користувачем, а також прикріплення до реєстру зіпсованих бланків
dogovir	Договір страхування
my_users	Інформація про користувачів системи
ref_bso_companyid	Словник компаній
ref_bso_statusid	Словник статусів бланків
ref_bso_subdivisionid	Словник підрозділів
ref_dogovir_car_type	Словник типу ТЗ
ref_dogovir_car_zona	Словник зони реєстрації ТЗ
ref_dogovir_clienttype	Словник типу клієнта
ref_dogovir_date	Інформація про дати сторення, початок та кінець дії договору та страхового покриття
ref_dogovir_documents	Перелік можливих документів страхувальника та булеві значення полів
ref_dogovir_insuredbounty	Інформація про страхову премію по договору
ref_dogovir_insuredsum	Інформація про страхову суму по договору
ref_dogovir_object_car	Інформація про автомобіль страхувальника
ref_dogovir_person_persons	Інформація про страхувальника
ref_dogovir_status	Словник статусів договору

Після створення таблиць та налаштування прав доступу до них було створено первинні та зовнішні ключі, а також проведено індексацію деяких полів, по яким буде проводитись фільтрація даних.

Завершальним етапом створення БД є заповнення словників. За допомогою програми phpMyAdmin це робиться досить швидко. Для прикладу в таблиці 2 відображена таблиця типу реєстрації автомобіля, яка містить наступні атрибути:

- ID
- Name
- Description
- Value

Таблиця 4.2 —Приклад заповненого словника в БД

ID	Name	Description	Value
1	B1	до 1600	1
2	B2	від 1601 до 2000	1,14
3	B3	від 2001 до 3000	1,18
4	B4	понад 3001	1,82
5	B5	електромобілі	0,9
6	F	причіп легкові	0,34
7	D1	до 20 осіб	2,55
8	D2	більше 20 осіб	3
9	C1	до 2 тон	2
10	C2	понад 2 тони	2,18
11	E	причіп вантаж	0,5
12	A1	до 300 куб. см.	0,34
13	A2	більше 301 куб. см	0,68

## 4.4 Реалізація програмного продукту

### 4.4.1 Реалізація створення бланків

Доступ до створення бланків має лише одна особа. Під час завантаження сторінки в першу чергу перевіряється можливість створення бланків в системі за допомогою маркеру, який зберігається в РНР сесії. Сама перевірка та збереження маркерів відбувається на стороні серверу, отже клієнт не може самовільно змінити його. При внесенні бланків вони автоматично починають знаходитись на обліку у користувача «ПРЕСТИЖ Склад».

Для додавання бланків в систему потрібно ввести наступні дані:

- компанія, яка передала бланки;
- серія;
- мінімальне та максимальне значення номеру;
- дата отримання бланку.

Також можна додати коментар до створених бланків за потреби.

Враховуючи те, що всі БСЗ друкує МТСБУ – кожен бланк має унікальну пару серія-номер. При введенні в поля серії та мінімального та максимального номеру в БД перевіряється правильність введення даних:

- серія введена кирилицею;
- мінімальний номер менше максимального;
- бланки в введеному діапазоні відсутні в системі.

Якщо потрібно додати декілька бланків – всю інформацію треба ввести ще раз. Після додавання бланків це автоматично зберігається в логах.

Форма створення бланків зображена на рисунку 4.10.

Рисунок 4.10 – Форма створення бланків

#### 4.4.2 Реалізація передачі бланків

Для передачі бланків потрібно вносити серію, мінімальний та максимальний номер бланків. Якщо бланки мають однакову інформацію стосовно статусу та користувача, тоді з БД автоматично завантажиться наступна інформація:

- компанія, яка передала бланки;
- найменування продукту;
- статус бланку;
- поточний користувач бланку.

Якщо поточним користувачем бланків є «ПРЕСТИЖ Склад» - їх може взяти лише людина, яка відповідає за склад та добавляє і розподіляє бланки.

Якщо потрібно передати бланки різних статусів, або бланки, які мають різні серії, чи інший діапазон номерів – реалізована можливість додавання бланків.

При передачі потрібно вказати нового користувача. В поле «Акт» автоматично внесеться номер акту передачі акту, який складається з номеру підрозділу, унікального номеру користувача та порядковим номером акту. Також можливо

додати коментар до передачі актів.

Важливо розуміти, що користувач бачить лише бланки відповідно до ієрархії. Головний офіс має доступ до всіх бланків, однак агенти можуть передавати лише ті бланки, які знаходяться в них на обліку. Директори підрозділів бачать бланки всіх підпорядкованих осіб, однак звичайний співробітник може передати та подивитись інформацію лише по тим бланкам, які передані йому в системі.

Форма передачі бланків зображена на рисунку 4.11.

The screenshot shows a web browser window with the URL `pwa.prestige-ic.com.ua/prestige/blank/change/change.php`. The page title is 'Зміна бланку'. On the left, there is a sidebar with a user profile 'LShydenko' and a menu with items: 'Бланки', 'Створення', 'Передача' (selected), 'Акти передачі бланків', 'Картка бланку', 'Вигрузка бланків', 'Створення реєстру зіпсованих', 'Перегляд реєстру зіпсованих', 'Договір', 'Довідники', 'Інструкція', and a 'Вийти' button. The main content area contains the 'Передача' form with the following fields:

- Серія бланку:
- Мінімальне значення номера:
- Максимальне значення номера:
- Компанія:
- Найменування продукту:
- Статус:
- Користувач:
- Новий стан бланку:
- Новий користувач:
- Дата зміни:
- Акт:
- Коментар:

At the bottom of the form are two buttons: 'ЗБЕРЕГТИ' (Save) and 'ДОДАТИ БЛАНКИ' (Add Blanks).

Рисунок 4.11 – Форма передачі бланків

#### 4.4.3 Реалізація створення акту передачі бланків

При передачі бланків автоматично створюється запис в БД в таблиці «bso\_acts», а в таблиці логів помічається унікальний ідентифікатор акту при передачі. Користувач може в будь-який момент переглянути всі акти передачі. Для більш зручного користування була додана фільтрація за місяцем передачі та користувачем, який так чи інакше пов'язаний з передачею бланків. Особа може або володіти бланком, або сама передати його, або бланки можуть бути передані їй. Перелік актів передачі зображений на рисунку 4.12.

Акти передачі БСЗ

pwa.prestige-ic.com.ua/prestige/blank/acts/index.php

ЛShydenko

Користувач якому передали бланки: Всі

Місяць передачі: червень 2020 р.

Номер акту	Дата передачі	Користувач, якому передали бланки	Користувач, чий бланки передавали	Користувач, який передав бланки
101-15419-11	2020-06-03	Михайло Віталійович Шиденко	ІПРІ/ТІМ/Служба	Шиденко ЛІлія Володимирівна
100-00012-314	2020-06-03	Шиденко ЛІлія Володимирівна	Михайло Віталійович Шиденко	Шиденко ЛІлія Володимирівна
100-00012-313	2020-06-03	Шиденко ЛІлія Володимирівна	Михайло Віталійович Шиденко	Шиденко ЛІлія Володимирівна
101-14319-10	2020-06-02	СК "Альфа" (Партнерська Служба)	ІПРІ/ТІМ/Служба	Шиденко ЛІлія Володимирівна
101-10517-26	2020-06-02	Шиденко ЛІлія Володимирівна	ІПРІ/ТІМ/Служба	Шиденко ЛІлія Володимирівна
100-00012-312	2020-06-02	Шиденко ЛІлія Володимирівна	Шиденко ЛІлія Володимирівна	Шиденко ЛІлія Володимирівна
000-001-10	2020-06-02	ІПРІ/ТІМ/Служба	Шиденко ЛІлія Володимирівна	Шиденко ЛІлія Володимирівна
101-17219-6	2020-06-01	Шиденко ЛІлія Володимирівна	ІПРІ/ТІМ/Служба	Шиденко ЛІлія Володимирівна
070-00001-49	2020-06-01	Михайло Віталійович Шиденко	ІПРІ/ТІМ/Служба	Шиденко ЛІлія Володимирівна

Вийти

Рисунок 4.12 – Перелік актів передачі бланків

Користувач має можливість переглянути акт, завантаживши його DOC версію, та відкрити за допомогою, наприклад, Microsoft Word. Для цього необхідно натиснути на номер акту, який потрібно завантажити. Акт передачі складається в двох примірниках. На рисунку 4.13 зображена одна частина цього акту.

**АКТ № 101-15419-11 від 03.06.2020**  
**Передачі бланків суворої звітності**

Начальник відділу по роботі з партнерами та регіональною мережею ПрАТ «СК «ПРЕСТИЖ» Шиденко Лілія Володимирівна, з однієї сторони  
та  
агент Михайло Віталійович Шиденко з другої сторони, склали цей акт в тому, що перша сторона передала, а друга прийняла наступні бланки суворої звітності:

Найменування бланків	Серія	№№ бланків		Кількість книжок, шт.
		з №	по №	
2_о_ОСЦПВ (СК "Альфа-Гарант")	АО	5955196	5955202	7
2_о_ОСЦПВ (СК "Альфа-Гарант")	АО	7046963	7046969	7
<b>Всього</b>				<b>14</b>

Підписи сторін:

Передав \_\_\_\_\_/Шиденко Л.В./      Прийняв \_\_\_\_\_/Михайло В.Ш./

Рисунок 4.13 – Загальний вигляд акту передачі бланків

#### 4.4.4 Реалізація перегляду історії по бланку

Для перегляду історії по бланку створена сторінка «Картка бланку». Користувачу пропонується ввести серію та номер бланку. Після цього з БД автоматично завантажиться інформація стосовно компанії, що дала бланк, поточного стану, продукту та користувача БСЗ. Це зроблено для попередньої перевірки введеної інформації. По натисканню на кнопку «Знайти» в таблиці логів знаходяться всі зміни атрибутів конкретного бланку та виводяться в таблицю на сторінці. Цей процес показаний на рисунку 4.14.

Дата	Статус	Користувач	Коментар
17.04.2020	Чистий	ПРЕСТИЖ-Склад	
03.06.2020	Чистий	Користувач	

Рисунок 4.14 – Перегляд інформації по бланку

#### 4.4.5 Реалізація створення звіту за фільтрами

При проектуванні бази даних було закладено досить багато довідників та створено велику кількість індексів для швидкого пошуку, адже висока швидкість знаходження даних була однією з найважливіших вимог до системи [5].

При знаходженні бланків дається можливість відфільтрувати по наступним атрибутам:



- дата створення бланку в системі;
- серія бланків, або всі серії;
- діапазон номерів бланків;
- компанія, яка передала бланки;
- найменування продукту;
- статус бланків;
- поточний користувач бланків;
- підрозділ, в якому знаходиться бланки.

Також користувач може вибрати які дані будуть відображені. Зображення форми для знаходження БСЗ зображена на рисунку 4.15. Результат виконання пошуку можна відобразити або на екрані в новому вікні (рисунок 4.16), або у CSV файл (рисунок 4.17), який відкривається за допомогою програми «Microsoft Excel». При збереженні в файл скрипт автоматично додає до заголовку файлу «BOM», що означає що файл повинно відкривати з кодуванням Unicode. Інакше кирилиця не буде відображатись.

The screenshot shows a web browser window with the URL `pwa.prestige-ic.com.ua/prestige/blank/blankfilter/blankfilter.php`. The page title is 'Вигрузка бланків'. The interface is for a user named 'LSHydenko'. On the left sidebar, under 'Бланки', there are links for 'Створення', 'Передача', 'Акти передачі бланків', 'Картка бланку', and 'Вигрузка бланків'. The 'Вигрузка бланків' section is active, showing options for 'Створення реєстру зіпсованих' and 'Перегляд реєстру зіпсованих'. The main content area has search filters: 'Початок періоду' (dd.mm.yyyy), 'Кінець періоду' (dd.mm.yyyy), 'Серія бланку', 'Стан бланку' (dropdown), 'Компанія' (dropdown), 'Номер з' (input), 'Номер по' (input), 'Вибрати все' (checkbox), 'Дата' (checkbox), 'Серія' (checkbox), 'Номер' (checkbox), 'Користувач' (checkbox), 'Договір №' (checkbox), 'Статус' (checkbox), 'Підрозділ' (checkbox), 'Всі серії' (checkbox), 'Користувач' (dropdown), 'Підрозділ' (dropdown), 'Всі' (dropdown), 'Продукт' (checkbox), 'Компанія' (checkbox), and 'Коментар' (checkbox). At the bottom, there are buttons 'На екран' and 'У CSV'.

Рисунок 4.15 – Форма для знаходження БСЗ

Дата створення	Дата останньої зміни	Серія	Номер	Продукт	Компанія	Стан	Підрозділ	Користувач	Договір №	Коментар
15/04/20	03/06/20	АО	5955196	2_o_ОСЦПВ (n)	СК "Альфа-Гарант"	Чистий	Агенти	Blank		
15/04/20	03/06/20	АО	5955197	2_o_ОСЦПВ (n)	СК "Альфа-Гарант"	Чистий	Агенти	Blank		
15/04/20	03/06/20	АО	5955198	2_o_ОСЦПВ (n)	СК "Альфа-Гарант"	Чистий	Агенти	Blank		
15/04/20	03/06/20	АО	5955199	2_o_ОСЦПВ (n)	СК "Альфа-Гарант"	Чистий	Агенти	Blank		
15/04/20	03/06/20	АО	5955200	2_o_ОСЦПВ (n)	СК "Альфа-Гарант"	Чистий	Агенти	Blank		
15/04/20	03/06/20	АО	5955201	2_o_ОСЦПВ (n)	СК "Альфа-Гарант"	Чистий	Агенти	Blank		
15/04/20	03/06/20	АО	5955202	2_o_ОСЦПВ (n)	СК "Альфа-Гарант"	Чистий	Агенти	Blank		

Рисунок 4.16 – Відображення даних в новому вікні

Дата створення	Дата останньої зміни	Серія	Номер	Продукт	Компанія	Стан	Підрозділ	Користувач	Договір №	Коментар
15.04.2020	15.04.2020	АО	5955196	2_o_ОСЦПВ (n)	СК "Альфа-Гарант"	Чистий	ПРЕСТИЖ	ПРЕСТИЖ-Склад		
15.04.2020	15.04.2020	АО	5955197	2_o_ОСЦПВ (n)	СК "Альфа-Гарант"	Чистий	ПРЕСТИЖ	ПРЕСТИЖ-Склад		
15.04.2020	15.04.2020	АО	5955198	2_o_ОСЦПВ (n)	СК "Альфа-Гарант"	Чистий	ПРЕСТИЖ	ПРЕСТИЖ-Склад		
15.04.2020	15.04.2020	АО	5955199	2_o_ОСЦПВ (n)	СК "Альфа-Гарант"	Чистий	ПРЕСТИЖ	ПРЕСТИЖ-Склад		
15.04.2020	15.04.2020	АО	5955200	2_o_ОСЦПВ (n)	СК "Альфа-Гарант"	Чистий	ПРЕСТИЖ	ПРЕСТИЖ-Склад		
15.04.2020	15.04.2020	АО	5955201	2_o_ОСЦПВ (n)	СК "Альфа-Гарант"	Чистий	ПРЕСТИЖ	ПРЕСТИЖ-Склад		
15.04.2020	15.04.2020	АО	5955202	2_o_ОСЦПВ (n)	СК "Альфа-Гарант"	Чистий	ПРЕСТИЖ	ПРЕСТИЖ-Склад		

Рисунок 4.17 – Збереження даних в CSV файл

#### 4.4.6 Реалізація створення та перегляду реєстру зіпсованих бланків

Під час виписування договору співробітник може допустити помилку, і тоді бланк буде вважатись зіпсованим. Ці зіпсовані бланки повинні бути переданими компанії, яка видала цей бланк. Для цього на основі бланків створюють реєстри. При зміні статусу БСЗ або договору на «Зіпсований» – в таблицю «bso\_punched\_register\_blanks» автоматично додається унікальний ідентифікатор бланку. Одним з атрибутів запису є «RegisterId» - номер реєстру з таблиці «bso\_punched\_register». Якщо значення цього атрибуту дорівнює null – це означає що бланк ще не був включений до жодного реєстру, і він буде відображатись при створенні нового реєстру.

Перш за все користувачу потрібно вибрати компанію, для якої буде створений реєстр та діапазон дат псування БСЗ. Також слід ввести номер реєстру. Для додавання бланку в реєстр потрібно помітити checkbox в стан «включений», натисканням на нього. Форма створення реєстру зіпсованих бланків зображена на рисунку 4.18.

Формування реєстру зіпсованих

Номер реєстру:

Компанія:

Дата створення реєстру:

Дата псування бланку з:  по:

<input type="checkbox"/>	Бланк
<input type="checkbox"/>	АО-6871513
<input type="checkbox"/>	АО-6871615
<input type="checkbox"/>	АМ-7569267
<input type="checkbox"/>	АР-7094259
<input type="checkbox"/>	АР-7192913
<input type="checkbox"/>	АР-7192929
<input type="checkbox"/>	АР-7192931
<input type="checkbox"/>	АР-7192944
<input type="checkbox"/>	АР-7193807
<input type="checkbox"/>	АР-7193852

Всього бланків вибрано: 0

**СТВОРИТИ РЕЄСТР**

Рисунок 4.18 – Створення реєстру зіпсованих бланків

Для перегляду списку реєстрів зіпсованих бланків потрібно відкрити сторінку «Перегляд реєстру зіпсованих». Загальний вигляд сторінки зображений на рисунку 4.19. В таблиці рядки помічені різними кольорами. Таким чином позначається статус реєстру. Якщо колір рядка зелений – реєстр був переданий компанії. В іншому випадку бланки знаходяться в офісі та будуть передані в найближчий час.

По натисканню на номер реєстру (рисунок 4.20) користувачу відображається перелік бланків та дається можливість ввести дату передачі. Також присутня функція експорту реєстру в CSV файл. Результат експорту зображений на рисунку 4.21.

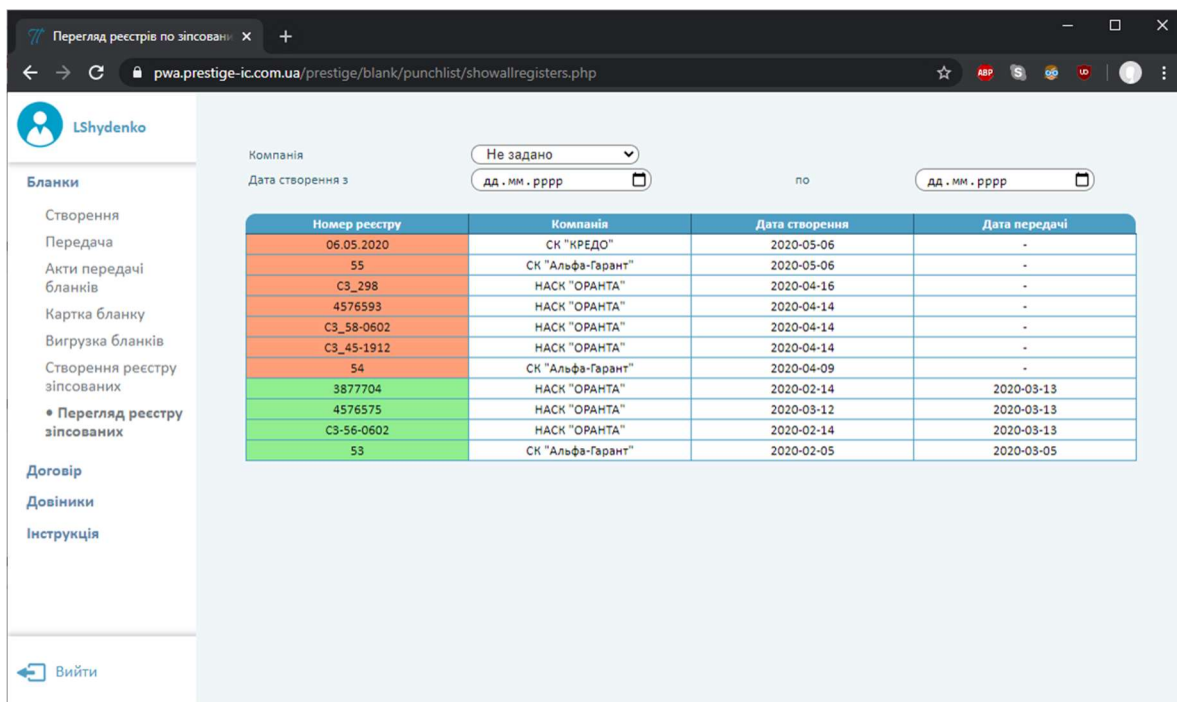


Рисунок 4.19 – Перегляд списку реєстрів зіпсованих бланків

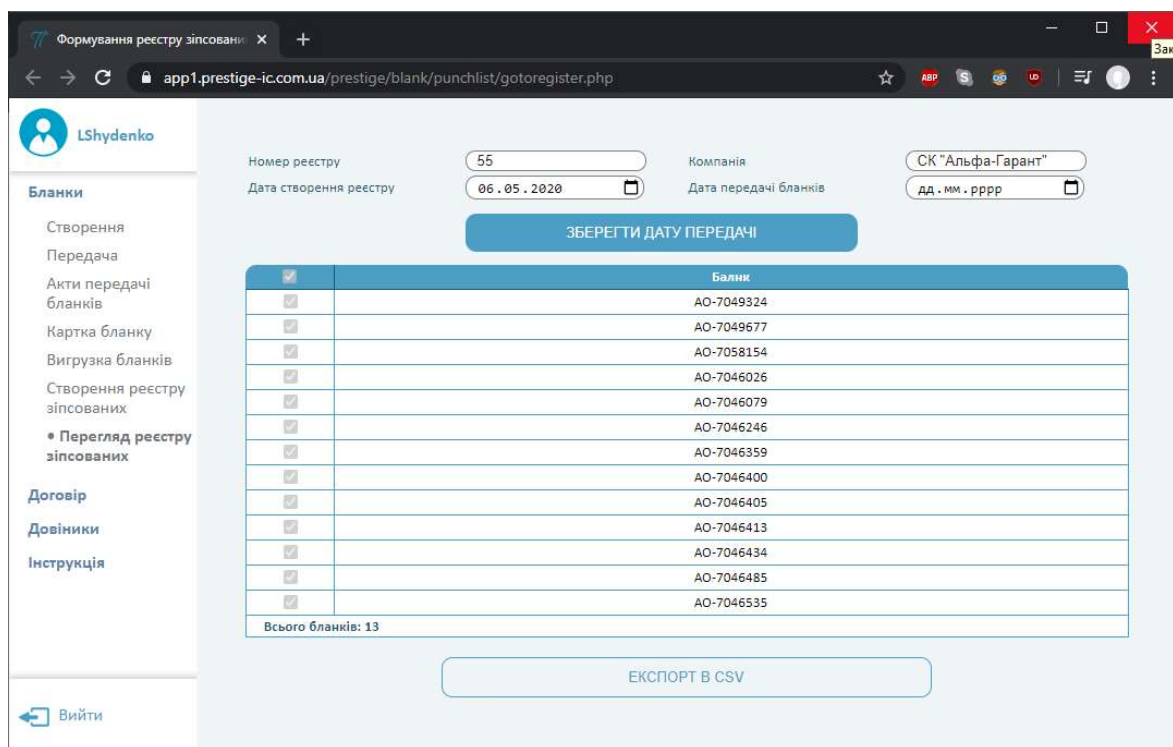


Рисунок 4.20 – Перегляд реєстру зіпсованих бланків

	A	B	C	D
1	Реєстр зіпсованих бланків № 55			
2	Серія	Номер		
3	АО	7049324		
4	АО	7049677		
5	АО	7058154		
6	АО	7046026		
7	АО	7046079		
8	АО	7046246		
9	АО	7046359		
10	АО	7046400		
11	АО	7046405		
12	АО	7046413		
13	АО	7046434		
14	АО	7046485		
15	АО	7046535		
16	Всього	13		
17				

Рисунок 4.21 – Перегляд експорту реєстру зіпсованих бланків в CSV

#### 4.4.7 Реалізація створення договору страхування

Для створення договору страхування використовують БСЗ. В системі розроблена окрема сторінка для створення договорів (рисунок 4.22). На ній користувачу потрібно ввести всі дані стосовно його наповнення. Створення договору починається з вибору продукту та бланку. В нашому випадку продукт – ОСЦПВ. Наступний етап – внесення серії та номеру бланку, після якого з БД автоматично завантажиться інформація стосовно компанії, та користувача.

Створення договору

Договір № АО - 1373299

Дублікат № 1373299

Компанія СК "КРЕДО"

Найменування продукту 2\_о\_ОСЦПВ (п)

Серія бланку АО

Номер бланку 1373299

Користувач Шиденко Лілія Володимирівна

Канал Офісні продажі

Реєстр агента

РДПП

Оригінал договору В роботі

Договір перевірений

Стан бланку Чистий

Стан договору Чернетка

Коментар

<< Розгорнути все >>

>> Згорнути все <<

Дати

Вийти

Рисунок 4.22 – Створення договору в системі

Можливі ситуації, коли користувач вводить номер бланку чи серії не правильно, або бланк знаходиться в стані використаного або зіпсованого. В такому випадку користувачу виводиться повідомлення про це, адже на їх основі не можна створити договір. Також є перевірка можливості використання бланку відповідно до ієрархії в СК.

Наступний етап створення договору – введення дати дії та укладання договору. При цьому відбувається чотири перевірки:

- дата укладання договору не більша за сьогоднішню;
- з моменту укладання договору до внесення в систему пройшло не більше трьох днів;
- дата дії договору не більша за рік;
- дата початку дії договору раніше за дату закінчення дії.

При цьому в окремому полі відображається строк дії договору в днях.

Страхова сума за договором виставляється автоматично та не може бути змінена, адже вона визначена МТСБУ в сумі 390 000 грн. Сума франшизи відрізняється в залежності від компанії, що надала бланк.

Заповнення дат та страхової суми зображено на рисунку 4.23.

Рисунок 4.23 – Заповнення дат та страхової суми



Користувачу також потрібно ввести інформацію стосовно страхувальника. Блок внесення цих даних зображений на рисунку 4.24.

The screenshot shows a web browser window with the URL `pwa.prestige-ic.com.ua/prestige/dogovor/vedennyadogovoru.php`. The page title is 'Створення договору' (Contract Creation). On the left, there is a sidebar with a user profile 'LSHydenko' and a menu with items: 'Бланки' (Forms), 'Договір' (Contract), 'Створення договору' (Contract Creation), 'Знайти договір' (Find contract), 'Вигрузка договорів' (Download contracts), 'Довідники' (Reference), and 'Інструкція' (Instructions). The main content area is titled 'Страхувальник' (Insured). It contains the following fields:

- Тип страхувальника (Insured type):  (Physical person)
- Ознака резидентності (Residency indicator):  (Is a resident)
- ІПН / ЄДРПОУ страхувальника (Tax ID / VAT ID of the insured):
- ПІБ / Найменування страхувальника (Full name / Name of the insured):
- Банківські реквізити (Bank details):
- Адреса страхувальника (Address of the insured):
- Дата народження страхувальника (Date of birth of the insured):
- Документ, що посвідчує особу страхувальника (Document confirming the identity of the insured):  (Passport)
- Серія: (Series):
- Номер: (Number):
- Дата видачі: (Date of issue):
- Ким виданий документ: (Issued by):
- Телефон страхувальника (Phone of the insured):
- Електронна пошта страхувальника (Email of the insured):

At the bottom of the form, it says 'Об'єкт страхування - Авто' (Insured object - Car). There is a 'Вийти' (Exit) button in the bottom left corner and a green arrow icon in the bottom right corner.

Рисунок 4.24 – Заповнення інформації щодо страхувальника

При цьому при внесенні ІПН відбувається його перевірка на правильність. Для цього потрібно:

1. знайти контрольну суму з наступними ваговими коефіцієнтами 2, 4, 10, 3, 5, 9, 4, 6, 8, 0;
2. знайти остачу від ділення контрольної суми на 11;
3. результат в пункті 2 повинен дорівнювати останньому числу ІПН;
4. якщо результат більший за 9 – треба знайти остачу від ділення контрольної суми на 10.

Також відбувається перевірка правильності введення дати народження. Перші 5 цифр в індивідуальному податковому номері – кількість днів від 01.01.1900 до дати народження страхувальника. Якщо ці дані не співпадають – система сповістить користувача про це.

Страховий платіж залежить від умов договору. Кожна компанія сама визначає коефіцієнти. Зазвичай на страховий платіж впливають:

- термін дії договору;
- тип страхувальника;
- тип транспортного засобу;
- кількість місяців користування транспортним засобом;
- місце реєстрації транспортного засобу;
- використання авто, як таксі;
- обов'язковість проходження контролю;
- вибрана сума франшизи.

Після заповнення всіх вищевказаних даних користувач системи побачить суму страхового платежу (рисунк 4.25).

**Створення договору**

Номер:

Дата видачі:

Ким виданий документ:

Телефон страхувальника:

Електронна пошта страхувальника:

---

### Об'єкт страхування - Авто

Тип ТЗ:

Державний номер ТЗ:

Марка та модель ТЗ:

Рік випуску авто:

Номер кузова (VIN-код):

Місце реєстрації ТЗ:

- ТЗ використовується протягом календарних місяців: ☒ (позначити) ☐ (не позначати)
- ТЗ використовується як таксі/маршрутне таксі: ☐ (позначити) ☒ (не позначати)
- ТЗ підлягає обов'язковому технічному контролю(ОТК): ☐ (позначити) ☒ (не позначати)
- до керування допущені особи з водійським стажем менше 3-х років та / або водійський стаж страхувальника менше 3-х років: ☐ (позначити) ☒ (не позначати)

(для річних договорів) 1 2 3 4 5 6 7 8 9 10 11 12  
☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒ ☒

Дата наступного ОТК:

Бп	K1	K2	K3	K4	K5	K6	K(строк)	K(бонус-малус)	K(зменшувальний)	Зменшення пільговику(%)
180	1.00	3.5	1.0	1.6	1.00	1.20	1	1	1	1

Страховий платіж:

[Вийти](#)

Рисунок 4.25 – Заповнення інформації щодо авто та розрахунок вартості страхового платежу

Після внесення цих даних та натискання кнопки «СТВОРИТИ», договір автоматично створюється, а бланк змінює свій статус на використаний.



#### 4.4.8 Реалізація пошуку договору страхування

Для пошуку договору страховики зазвичай використовують ІПН або прізвище страхувальника. На рисунку 4.26 зображено приклад одночасного використання пошуку по частині ІПН (456) та з входженням імені «Вадим» в ПІБ.

#	ІПН	В ролі	ПІБ	Договір
1	456	Страхувальник	Вадим	АО - 7049315

Рисунок 4.26 – Приклад пошуку договору за ІПН та ПІБ

Після знаходження номеру договору достатньо натиснути на нього двічі, і користувач зможе переглянути всю інформацію за договором.

У випадку коли відомий номер договору, користувачу достатньо ввести його в відповідне поле форми та натиснути на кнопку «знайти» під цим полем.

Після натискання на кнопку або на номер договору в таблиці система знаходить унікальний ідентифікатор договору та знаходить всі потрібні дані з таблиць та словників, після чого завантажується сторінка (рисунок 4.27) з відображенням цих даних. Для зручності користувача у верхній частині екрану завжди показується номер договору та його стан.

Перегляд договору x +

pwa.prestige-ic.com.ua/prestige/dogovor/vedennyadogovoru.php

Ви в договорі АО - 5929744, укладений

Договір № АО - 5929744

Компанія СК "Альфа-Гарант"

Найменування продукту 2\_о\_ОСЦПВ (п)

Серія бланку АО

Користувач Канал

Реєстр агента РДПП

Оригінал договору В архіві

Стан бланку Передано в СК

Коментар

Для номеру використовується бланк

5929744

Договір перевірений

Укладений

<< Розгорнути все >>

>> Згорнути все <<

Дати

Дата створення договору 13.12.2019

Початок дії 14.12.2019

Строк дії договору в днях 365

Закінчення дії 13.12.2020

Страхова сума

Загальна страхова сума за договором 390 000.00

Валюта страхової суми UAH

Вийти

Рисунок 4.27 – Приклад відображення даних по договору

#### 4.4.9 Реалізація друку договору страхування та пов'язаних документів

Для друку договору перш за все потрібно перейти в сам договір, та внизу сторінки натиснути кнопку «друк». Після натискання автоматично відкриється нове вікно з згенерованим договором страхування. Для створення PDF використовується бібліотека TCPDF. Скріншот такого файлу зображений на рисунку 4.28.

Також у користувача є можливість створити рахунок-фактуру. Скріншот цього файлу зображений на рисунку 4.29.

0 0 0 0 1 1 1 0 1 8 1 0 1 0 1 9

1000,00 (одна тисяча гривень, 00 копійок)

2 3 0 2 1 9 7 7

Посвідчення водія

ДІА ГУ МВС України в м. Києві

PM3 81011

Київ

Ні  
Ні  
Ні

180 0.34 4.80 1.00 1.50 1.00 0.00 1.00 1.00 1.00 1.00

3 9 6 0 0

триста дев'яносто шість гривень, 58 копійок

0 0 0 0 1 0 1 0 1 8

1 0 1 0 1 8

Рисунок 4.28 – Скріншот згенерованого PDF для друку договору

**Постачальник** ПрАТ "СК "ПРЕСТИЖ"  
п/р UA603220010000026506001090518, в АТ "Універсал Банк"  
ЄДРПОУ 38272117

**Платник**

**Рахунок-фактура № AM-5496481**  
від 10.10.2018 р.

№	Призначення платежу	Страховий платіж, грн.
1	Страховий платіж за договором страхування № AM-5496481 від 10.10.2018р. в сумі 396,58 грн. без ПДВ, клієнт	396,58
<b>Разом:</b>		<b>396,58</b>

Сума прописом: Триста дев'яносто шість грн., 58 коп., без ПДВ

Рахунок виписав (ла): \_\_\_\_\_

Рисунок 4.29 – Скріншот рахунку-фактури

#### 4.4.10 Реалізація створення звіту по договорах страхування

Як у випадку з бланками, у користувача є можливість створити звіт по договорам страхування. На рисунку 4.30 зображено форму для вибору фільтрів та даних, що будуть завантажуватись. У користувача також є два варіанти відображення звіту: на екран та у CSV.

Рисунок 4.30 – Форма для створення звіту по договорах

## 5 ВИСНОВОК

Під час виконання дипломної роботи були досягнуті наступні цілі:

1. Спроектовано систему з документообігу в страховій компанії;
2. Завдяки VPN мережі реалізовано можливість відключитись до спроектованої системи з будь-якої точки світу;
3. Впроваджено алгоритм авторизації та аутентифікації користувачів з визначенням ролей та можливостей користувачів;
4. Створено базу даних, заповнені всі потрібні довідники;
5. Розроблено алгоритм контролю за бланками суворої звітності з збереженням всіх змін, які робляться з ними;
6. Налаштовано автоматичне створення актів передачі бланків;
7. Реалізовано функцію створення договорів страхування на основі бланків суворої звітності;
8. Створено можливість перевірки ІПН та пошуку інформації у базі даних за вказаним ІПН;
9. Розроблено та реалізовано алгоритм пошуку договору за номером, ІПН та ПІБ страхувальника;
10. Створено можливість формування звітів з використанням різноманітних фільтрів;
11. Реалізовано можливість друку договору на бланку суворої звітності з системи за допомогою PDF файлу;
12. Розроблена можливість формування рахунку-фактури з системи;
13. Використана кольорова гама підприємства – замовника
14. Станом на кінець травня в систему було внесено більше 27 000 бланків суворої звітності та створено більше 11 000 договорів.

## СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Антіпов В., Законодавство України про страхування // К.: Юрінком, 1997. – № 4. – 368 с.
2. Базилевич В., Страхова справа. – К.: Знання, 1997., 1997. – 216 с
3. Базилевич В., Страховий ринок України. – К.: Товариство “Знання”, 1998. – 374 с.
4. Браун Е., Изучаем JavaScript. Руководство по созданию современных веб-сайтов. — К.: тов-во “Діалектика”, 2020. — 368 с.
5. Бьюли А., Изучаем SQL. — СПб.: ООО “Символ”, 2007. — 308 с.
6. Гайдаржи В., Изварин И. Базы данных в информационных системах. — К.: Университет "Україна", 2018. — 418 с.
7. Дакетт Д., HTML и CSS. Разработка и дизайн веб-сайтов — М.: Эксмо, 2013. — 480 с.
8. Дамашке Г., PHP и MySQL. — М.: “НТ Пресс”, 2012. — 320 с.
9. Заруба О., Страхова Страхова справа: підручник підручник. — К.: тов-во „Знання”, 1998., 1998. – 321 с.
10. Кофлер М. Linux. Установка, настройка, администрирование. — СПб.: Питер, 2014. — 768 с.
11. Мартин Р., Чиста архітектура. — Х.: “Фабула”, 2019. — 368 с.
12. Прищепа Є., Тимошченко А. Сервер на основі операційної системи FreeBSD 6.1. — К.: Университет "Україна", 2015. — 216 с.
13. Скляр Д. Изучаем PHP 7: руководство по созданию интерактивных веб-сайтов.: Пер. с англ. — СПб.: ООО “Альфа-книга”, 2017. — 464 с.
14. Стейнер К., Тотальна автоматизація. Як комп'ютерні алгоритми змінюють світ. — К.: “ Наш Формат”, 2018. — 280 с.
15. Титтел Е., Минник К. HTML5 и CSS3 для чайников. — К.: тов-во “Діалектика”, 2016. — 400 с.
16. Ульман Д., Хопкрофт Д., Ахо А. Структуры данных и алгоритмы. — К.: тов-во “Діалектика”, 2016. — 400 с.

17. Флэнаган Д. JavaScript. Подробное руководство. — СПб.: ООО “Символ”, 2018. — 1080 с.
18. Звіт нацкомфінпослуг за 2018 рік [Електронний ресурс] / – Режим доступу до ресурсу: <https://www.nfp.gov.ua/files/ZVIT/Публічний%20звіт%202018%20-%20на%20сайт.pdf>
19. УКАЗ ПРЕЗИДЕНТА УКРАЇНИ №133/2017 [Електронний ресурс] / – Режим доступу до ресурсу: <https://www.president.gov.ua/documents/1332017-21850>
20. Как я умудрился за 1 день задолжать Amazon 12000\$ [Електронний ресурс] / – Режим доступу до ресурсу: <https://habr.com/ru/post/357764/>

## ДОДАТОК А

Реалізація програмного рішення інформаційної системи  
з документообігу в страховій компанії

Специфікація

УКР.НТУУ”КПІ ім. Ігоря Сікорського”\_ТЕФ\_АПЕПС\_ТМ61

Аркушів 2

Київ – 2020



Позначення	Найменування	Примітки
Документація		
УКР.НТУУ«КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕ ПС_ТМ	Записка.docx	Текстова частина дипломної роботи
Компоненти		
УКР.НТУУ«КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕ ПС_ТМ	prestige.sql	База даних
УКР.НТУУ«КПІ ім. Ігоря Сікорського»_ТЕФ_АПЕ ПС_ТМ	prestige.zip	Архів з програмним кодом

## ДОДАТОК Б

Реалізація програмного рішення інформаційної системи  
з документообігу в страховій компанії

Текст програми

УКР.НТУУ «КПІ ім. Ігоря Сікорського»\_ТЕФ\_АПЕПС\_ТМ61

Аркушів 13

Київ – 2020

## Текст підпрограми для авторизації в системі

```
<?php
    session_start(); // Початок сесії
    $username = $_POST["form_username"];
    $_SESSION['username'] = $_POST["form_username"];
    $_SESSION['userpassword'] = $_POST["form_password"];
    $_SESSION['dbip'] = "127.0.0.1";
    $_SESSION['dbname'] = "prestige";
    if($link = mysqli_connect($_SESSION['dbip'], $_SESSION['username'], $_SESSION['userpassword'],
$_SESSION['dbname'])){ // спроба підключення до БД
        $r = mysqli_fetch_array(mysqli_query($link, "SELECT `id`, `Ierarhic`, `subdivisionid`
from `my_users` where `username` = '". $username ."'")); // збереження необхідних даних
        $_SESSION['userid'] = $r[0];
        $_SESSION['userierarhic'] = $r[1];
        $_SESSION['usersd'] = $r[2];
        mysqli_close($link);
        $_SESSION['connected'] = true;
    }
    else{
        $_SESSION = array();
        session_destroy();
        session_start();
        $_SESSION['connected'] = false;
        $_SESSION['result'] = "error";
    }
    header ('Location: index.php'); // перенаправлення на потрібну сторінку
    exit();
?>
```

## Текст підпрограми для відображення сторінки створення бланку

```
<?php
    error_reporting(0);
    session_start();
    if($_SESSION['connected'] == false){
        header ('Location: /prestige/Login.php'); // перенаправлення на потрібну сторінку
        exit();
    }
    else{
        $userid = $_SESSION['userid'];
        $link = mysqli_connect($_SESSION['dbip'], $_SESSION['username'],
$_SESSION['userpassword'], $_SESSION['dbname']);
        mysqli_set_charset($link, "utf8");
        $SD = mysqli_fetch_array(mysqli_query($link, "SELECT `subdivisionid` from `my_users`
where `id` = '". $userid ."'"));
        if(mysqli_fetch_array(mysqli_query($link, "SELECT `AllowAddBlanks` from `roles` where
`UID` = '". $userid ."'"))[0] == 0){
            header ('Location: /prestige/index.php'); // перенаправлення на потрібну
            exit();
        }
    }
?>
<html>

<!DOCTYPE html>
<html lang="ru">
<head>
<link rel="shortcut icon" href="/prestige/Icon.ico" type="image/x-icon">
<link href="/prestige/css/main.css" rel="stylesheet" />
<link href="/prestige/css/css.css" rel="stylesheet" />
<link href="/prestige/css/defaultpage.css" rel="stylesheet" />
<meta charset="utf-8">
<title>Створення бланку</title>
<script>
    function tryToCreate(){
        var AllFine = true;
        if(document.getElementById('form_company').value == 0 && AllFine == true){
            AllFine = false;
            alert("Перевірте введені данні (компанія)");
        }
        if(document.getElementById('form_product').value == 0 && AllFine == true){
            AllFine = false;
            alert("Перевірте введені данні (продукт)");
        }
    }
</script>
```

```

    }
    if(document.getElementById('form_date').value == "" && AllFine == true){
        AllFine = false;
        alert("Перевірте введені данні(дата)");
    }
    if(document.getElementById('form_user').value == 0 && AllFine == true){
        AllFine = false;
        alert("Перевірте введені данні(користувач)");
    }
    if(document.getElementById('form_seried').value == "" && AllFine == true){
        AllFine = false;
        alert("Перевірте введені данні(серія)");
    }
    if(document.getElementById('form_minnumber').value == "" && AllFine == true){
        AllFine = false;
        alert("Перевірте введені данні(мінімальний номер)");
    }
    if(document.getElementById('form_maxnumber').value == "" && AllFine == true){
        AllFine = false;
        alert("Перевірте введені данні(максимальний номер)");
    }
    if(document.getElementById('form_minnumber').value != "" &&
document.getElementById('form_maxnumber').value != "" && AllFine == true){
        var min = document.getElementById('form_minnumber').value;
        var max = document.getElementById('form_maxnumber').value;
        if(min <= max){
            var xhttp;
            if (min == "" || max == "") {
                return;
            }
            xhttp = new XMLHttpRequest();
            xhttp.onreadystatechange = function() {
                if (this.readyState == 4 && this.status == 200) {
                    if(this.responseText != 0){
                        alert("В вказаному діапазоні є існуючі
бланки!");
                    }
                    else{
                        if(confirm("Ви дійсно хочете створити
" +(max-min+1)+ " бланків?"))
                            document.getElementById('confirm').click();
                    }
                }
            };
            xhttp.open("GET",
"findblankinrange.php?s="+document.getElementById('form_seried').value+"&min="+min+"&max="+max, true);
            xhttp.send();
        }
        else {
            AllFine = false;
            alert("Перевірте введені данні(Мінімальний більше максимального)");
        }
    }
}
</script>
</head>
<body>
<script src = "/prestige/Menu.php?name=?php echo $username ?>"></script>
<?php
    if($_SESSION['result']) {
        echo('<div class = "block3" ondblclick = "this.remove();"><center>');
        echo $_SESSION['result'];
        $_SESSION['result'] = NULL;
        $_SESSION['ispositive'] = NULL;
        echo('</center></div>');
    }
?>
<div class = "block2">
<form method='post' action="insertaction.php" id="my_form"></form>
<table class = "maintable">
<tr class = "trtitle">
    <td style = "width: 50%;">
        Компанія
    </td>
    <td style = "width: 50%;">
        Найменування продукту

```

```

        </td>
</tr>
<tr class = "trinput">
    <td>
        <select id="form_company" name="form_company" form="my_form">
            <option value = 0 disabled selected>Виберіть компанію</option>
            <?php
                $result = mysqli_query($link, 'SELECT CompanyID, CompanyName FROM ref_bso_companyid');
                while($row = mysqli_fetch_array($result)){
                    echo '<option value = "'. $row[0].'">'. $row[1]. '</option>';
                }
            ?>
        </select>
    </td>
    <td>
        <select id="form_product" name="form_product" form="my_form">
            <option value = 0 disabled selected>Виберіть продукт</option>
            <?php
                $result = mysqli_query($link, 'SELECT ID, productid, simpleDescription FROM
ref_bso_productid where blank = 1 order by `productid`');
                while($row = mysqli_fetch_array($result)){
                    echo '<option value = "'. $row[0].'">'. $row[2]. '</option>';
                }
            ?>
        </select>
    </td>
</tr>
<tr class = "trtitle">
    <td>
        Дата створення бланку
    </td>
    <td>
        Користувач
    </td>
</tr>
<tr class = "trinput">
    <td>
        <input type="date" value="<?php echo date("Y-m-d");?>" id="form_date" name="form_date"
form="my_form">
    </td>
    <td>
        <select id = "form_user" name = "form_user" form="my_form">
            <?php
                $result = mysqli_query($link, 'SELECT id, names FROM my_users where
`id` = 1');
                while($row = mysqli_fetch_array($result)){
                    echo '<option value = "'. $row[0].'">'. $row[1]. '</option>';
                }
            ?>
        </select>
    </td>
</tr>
<tr class = "trtitle">
    <td>
        Серія бланку
    </td>
</tr>
<tr class = "trinput">
    <td>
        <input type="text" id="form_seried" name="form_seried" pattern="[0-9A-Яa-яЁё]+$" <?PHP
        if($_SESSION['seried']){ echo (' value = "'.$_SESSION['seried'].'" ');
$_SESSION['seried'] = null;}?> form="my_form">
    </td>
</tr>
<tr class = "trtitle">
    <td>
        Мінімальне значення номера
    </td>
    <td>
        Максимальне значення номера
    </td>
</tr>
<tr class = "trinput">
    <td>
        <input type="text" id="form_minnumber" name="form_minnumber" pattern="[0-9]+$" <?PHP
        if($_SESSION['num1']){ echo (' value = "'.$_SESSION['num1'].'" '); $_SESSION['num1'] = null;}?> required
form="my_form">
    </td>

```

```

        <td>
            <input type="text" id="form_maxnumber" name="form_maxnumber" pattern="[0-9]+$" <?PHP
if($_SESSION['num2']){ echo (' value = "'.$_SESSION['num2'].'" '); $_SESSION['num2'] = null;}> required
form="my_form">
        </td>
    </tr>
    <tr class = "trtitle">
        <td>
            Коментар
        </td>
    </tr>
    <tr class = "trinput">
        <td colspan = '2'>
            <input type = "text" <?php if($_SESSION['com']){echo'value = "'.$_SESSION['com'].'"';
$_SESSION['com'] = NULL;}> id="form_comment" name="form_comment" form="my_form">
        </td>
    </tr>
    <tr class = "trinput">
        <td>
            <input type="submit" id = "confirm" form="my_form" hidden>
            <input type="button" class = "submitbutton" value="СТВОРИТИ" onclick = "tryToCreate()">
        </td>
    </tr>
</table>
</div>
<script>
    displ('Blanks');
    document.getElementById('CreateBlankMenu').style.fontWeight = 600;
    document.getElementById('CreateBlankMenu').innerHTML = "&#149;";
    "+document.getElementById('CreateBlankMenu').innerHTML;
</script>
<body>
</html>

```

## Текст підпрограми для створення бланку в системі

```

<?php
    error_reporting(0);
    session_start();
    if($_SESSION['connected'] == false){
        header ('Location: /prestige/Login.php'); // перенаправлення на потрібну сторінку
        exit();
    }
    else{
        $userid = $_SESSION['userid'];
        $link = mysqli_connect($_SESSION['dbip'], $_SESSION['username'],
$_SESSION['userpassword'], $_SESSION['dbname']);
        mysqli_set_charset($link, "utf8");
        $SD = mysqli_fetch_array(mysqli_query($link, "SELECT `subdivisionid` from `my_users`
where `id` = ".$_userid))[0];
        mb_internal_encoding("UTF-8");
    }

    function exeptionfound($thislink){
        echo mysqli_error($thislink);
        mysqli_rollback($thislink);
        die();
    }

    $form_seried = $_POST["form_seried"];
    if($form_seried){$form_seried = mb_strtoupper( $form_seried, 'UTF-8' );}

    mysqli_autocommit($link,FALSE);
    $num = strlen((string)$_POST["form_minnumber"]);
    for($i = $_POST["form_minnumber"]; $i <= $_POST["form_maxnumber"]; $i++){
        $sql = 'INSERT INTO `bso`(`BSO_ID`, `BSO_Series`, `BSO_Number`, `BSO_StatusID`,
`BSO_DateOfCreation`, `BSO_CompanyID`, `BSO_UserID`, `BSO_ProductID`)
VALUES (NULL, "'.$form_seried.'", "'.$str_pad($i, $num, '0', STR_PAD_LEFT).'",
"1", " '.$_POST["form_date"].'", " '.$_POST["form_company"].'", " '.$_POST["form_user"].'",
"'.$_POST["form_product"].'")';
        mysqli_query($link, $sql) or exeptionfound($link);
        $sql2 = 'UPDATE bso_logs SET `comment` = "'.$_POST["form_comment"].'", `DateChanged` =
"'.$_POST["form_date"].'" ORDER BY id DESC LIMIT 1';
        mysqli_query($link, $sql2) or exeptionfound($link);
    }
    mysqli_commit($link);
    $_SESSION['ispositive'] = true;

```

```

        if($form_seried != "")
            $_SESSION['result'] = "Успішно створено " . ($_POST["form_maxnumber"] -
$_POST["form_minnumber"]+1) . " бланків серії " . $form_seried;
        else
            $_SESSION['result'] = "Успішно створено " . ($_POST["form_maxnumber"] -
$_POST["form_minnumber"]+1) . " бланків без серії";
        header ('Location: insert.php');
        exit();
    }
}
?>

```

## Текст підпрограми для звіту по бланкам на екран

```

<?php
    error_reporting(0);
    session_start();
    if($_SESSION['connected'] == false){
        header ('Location: /prestige/Login.php'); // перенаправление на нужную страницу
        exit();
    }
    else{
        $userid = $_SESSION['userid'];
        $link = mysqli_connect($_SESSION['dbip'], $_SESSION['username'],
$_SESSION['userpassword'], $_SESSION['dbname']);
        mysqli_set_charset($link, "utf8");
        $SP = $_SESSION['usersd'];
        $Ierarhic = $_SESSION['userierarhic'];
    }
?>
<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="utf-8">
<title>Вигрузка бланків</title>
<style>
body{
    margin: 0px;
}
#output {
    margin:0px;
    padding:0px;
    font-family: "Trebuchet MS", Arial, Helvetica, sans-serif;
    font-size: 12px;
    border-collapse: collapse;
    width: 100%;
}
#output td, #customers th {
    border-left: 1px solid #ddd;
    border-top: 1px solid #ddd;
    border-bottom: 1px solid #ddd;
    padding: 8px;
}
#output tr:nth-child(odd){background-color: AliceBlue;}
#output tr:hover {background-color: #b8e1ec;}
#output th {
    padding-top: 12px;
    padding-bottom: 12px;
    text-align: left;
    background-color: LightSeaGreen;
    color: white;
}
#output th:hover{
    cursor: pointer;
}
</style>
</head>
<?php
    date_default_timezone_set('Europe/Kiev');
    $form_date1 = new DateTime($_POST['form_date1']);
    $form_date2 = new DateTime($_POST['form_date2']);
    $form_date2->add(new DateInterval('P1D'));
    if($_POST["form_series"]){
        $form_series = $_POST["form_series"];
        $form_series = mb_strtoupper( $form_series, 'UTF-8' );
    }
    $_SESSION['form_number1'] = $_POST["form_number1"];
    $_SESSION['form_number2'] = $_POST["form_number2"];

```

```

$_SESSION['form_company'] = $_POST["form_company"];
$_SESSION['form_status'] = $_POST["form_status"];
$_SESSION['form_user'] = $_POST["form_user"];
$_SESSION['form_subdivision'] = $_POST["form_subdivision"];
$_SESSION['form_showdate'] = $_POST["form_showdate"];
$_SESSION['form_showseries'] = $_POST["form_showseries"];
$_SESSION['form_shownumber'] = $_POST["form_shownumber"];
$_SESSION['form_showproduct'] = $_POST["form_showproduct"];
$_SESSION['form_showcompany'] = $_POST["form_showcompany"];
$_SESSION['form_showstatus'] = $_POST["form_showstatus"];
$_SESSION['form_showsubdivision'] = $_POST["form_showsubdivision"];
$_SESSION['form_showsalepoint'] = $_POST["form_showsalepoint"];
$_SESSION['form_showuser'] = $_POST["form_showuser"];
$_SESSION['form_showcontract'] = $_POST["form_showcontract"];
$_SESSION['form_showcomment'] = $_POST["form_showcomment"];
// Формирование запроса, выбор что выводить
$query = "SELECT ";
$firstfilter = true;
if($_SESSION['form_showdate']){ if($firstfilter) {$firstfilter = false; }
else { $query = $query.','; } $query = $query.(select DATE_FORMAT(BSO_DateOfCreation,"%d/%m/%Y"),
(select DATE_FORMAT(bso_logs.`DateChanged`, "%d/%m/%Y") from bso_logs where bso_logs.`bso_id` =
`bso`.`BSO_ID` order by `DateChanged` DESC limit 1)); }
if($_SESSION['form_showseries']){ if($firstfilter) {$firstfilter = false; }
else { $query = $query.','; } $query = $query.`bso.bso_series`; }
if($_SESSION['form_shownumber']){ if($firstfilter) {$firstfilter = false; }
else { $query = $query.','; } $query = $query.`bso.bso_number`; }
if($_SESSION['form_showproduct']){ if($firstfilter) {$firstfilter = false; }
else { $query = $query.','; } $query = $query.(select ref_bso_productid.SimpleDescription from
ref_bso_productid where id = bso.BSO_ProductID); }
if($_SESSION['form_showcompany']){ if($firstfilter) {$firstfilter = false; }
else { $query = $query.','; } $query = $query.(select ref_bso_companyid.companyname from
ref_bso_companyid where companyid = bso.BSO_CompanyID); }
if($_SESSION['form_showstatus']){ if($firstfilter) {$firstfilter = false; }
else { $query = $query.','; } $query = $query.(select ref_bso_statusid.StatusDescription from
ref_bso_statusid where statusid = bso.BSO_StatusID); }
if($_SESSION['form_showsubdivision']){ if($firstfilter) {$firstfilter =
false; } else { $query = $query.','; } $query = $query.(SELECT
ref_bso_subdivisionid.SubdivisionDescription FROM ref_bso_subdivisionid where
ref_bso_subdivisionid.SubdivisionID = (select my_users.SubdivisionID from my_users where id =
`bso`.`BSO_UserID`)); }
if($_SESSION['form_showsalepoint']){ if($firstfilter) {$firstfilter = false; }
else { $query = $query.','; } $query = $query.(SELECT ref_bso_salepointid.SalePointDescription FROM
ref_bso_salepointid where ref_bso_salepointid.id = (select my_users.SalePointID from my_users where id =
`bso`.`BSO_UserID`)); }
if($_SESSION['form_showuser']){ if($firstfilter) {$firstfilter = false; }
else { $query = $query.','; } $query = $query.(select my_users.names from my_users where id =
`bso`.`BSO_UserID`); }
if($_SESSION['form_showcontract']){ if($firstfilter) {$firstfilter = false; }
else { $query = $query.','; } $query = $query.(REPLACE((select `dogovir`.`contract` from `dogovir` where
`dogovir`.`BlankID` = `bso`.`bso_id`), " ", "")); }
if($_SESSION['form_showcomment']){ if($firstfilter) {$firstfilter = false; }
else { $query = $query.','; } $query = $query.(select bso_logs.`Comment` from bso_logs where
bso_logs.`bso_id` = `bso`.`BSO_ID` order by `DateChanged` DESC, bso_logs.ID DESC limit 1); }
$query = $query.' from bso ';
$firstwhere = true;
if(isset($_POST['form_date1']) && $_POST['form_date1'] != ""){
    if($firstwhere){
        $query = $query." WHERE ";
        $firstwhere = false;
    }
    else{
        $query = $query." AND ";
    }
    $query = $query.' (bso.BSO_DateOfCreation >= "'. $form_date1-
>format('Y-m-d').'")';
}
if(isset($_POST['form_date2']) && $_POST['form_date2'] != ""){
    if($firstwhere){
        $query = $query." WHERE ";
        $firstwhere = false;
    }
    else{
        $query = $query." AND ";
    }
    $query = $query.' (`bso`.`BSO_DateOfCreation` <= "'. $form_date2-
>format('Y-m-d').'")';
}
}

```



```

if(isset($_POST['form_allseries']) == false){
    if($form_series){
        if($firstwhere){
            $query = $query." WHERE ";
            $firstwhere = false;
        }
        else{
            $query = $query." AND ";
        }
        $query = $query.' (bso.bso_series = "'.$form_series.'")';
    }
    else{
        if($firstwhere){
            $query = $query." WHERE ";
            $firstwhere = false;
        }
        else{
            $query = $query." AND ";
        }
        $query = $query.' (bso.bso_series = "")';
    }
}
if(isset($_SESSION['form_number1']) && $_SESSION['form_number1'] != ""){
    if($firstwhere){
        $query = $query." WHERE ";
        $firstwhere = false;
    }
    else{
        $query = $query." AND ";
    }
    $query = $query.' (bso.bso_number >=
'.$_SESSION['form_number1'].'")';
}
if(isset($_SESSION['form_number2']) && $_SESSION['form_number2'] != ""){
    if($firstwhere){
        $query = $query." WHERE ";
        $firstwhere = false;
    }
    else{
        $query = $query." AND ";
    }
    $query = $query.' (bso.bso_number <=
'.$_SESSION['form_number2'].'")';
}
if(isset($_SESSION['form_company']) && $_SESSION['form_company'] != 0){
    if($firstwhere){
        $query = $query." WHERE ";
        $firstwhere = false;
    }
    else{
        $query = $query." AND ";
    }
    $query = $query.' (bso.BSO_CompanyID =
'.$_SESSION['form_company'].'')';
}
if(isset($_SESSION['form_user']) && $_SESSION['form_user'] != 0){
    if($firstwhere){
        $query = $query." WHERE ";
        $firstwhere = false;
    }
    else{
        $query = $query." AND ";
    }
    $query = $query.' (bso.BSO_UserID = '.$_SESSION['form_user'].'')';
}
if(isset($_SESSION['form_status']) && $_SESSION['form_status'] != 0){
    if($firstwhere){
        $query = $query." WHERE ";
        $firstwhere = false;
    }
    else{
        $query = $query." AND ";
    }
    $query = $query.' (bso.BSO_StatusID =
'.$_SESSION['form_status'].'')';
}

```

```

    }
    if(isset($_SESSION['form_subdivision']) && $_SESSION['form_subdivision'] !=
0){
        if($firstwhere){
            $query = $query." WHERE ";
            $firstwhere = false;
        }
        else{
            $query = $query." AND ";
        }
        $query = $query.' ((select my_users.SubdivisionID from my_users
where id = bso.BSO_UserID) = ' .($_SESSION['form_subdivision'].')');
    }
    if($Ierarhic == 1){
        if($firstwhere){
            $query = $query." WHERE ";
            $firstwhere = false;
        }
        else{
            $query = $query." AND ";
        }
        $query = $query.' `BSO_userid` = '.$userid;
    }
    // Вывод результатов в таблицу
    $result = mysqli_query($link, $query) or die(mysqli_error($link));
    if(mysqli_num_rows($result) != 0){
        echo '<table id = "output">';
        echo '<thead>';
        if($_SESSION['form_showdate']){ echo '<th data-type="number" width =
1em>Дата створення</th><th data-type="number" width = 1em>Дата останньої зміни</th>'; }
        if($_SESSION['form_showseries']){ echo '<th data-type="string" width
= 1em>Серія</th>'; }
        if($_SESSION['form_shownumber']){ echo '<th data-type="number" width
= 2em>Номер</th>'; }
        if($_SESSION['form_showproduct']){ echo '<th data-type="string"
width = 2em>Продукт</th>'; }
        if($_SESSION['form_showcompany']){ echo '<th data-type="string"
width = 2em>Компанія</th>'; }
        if($_SESSION['form_showstatus']){ echo '<th data-type="string" width
= 2em>Стан</th>'; }
        if($_SESSION['form_showsubdivision']){ echo '<th data-type="string"
width = 2em>Підрозділ</th>'; }
        if($_SESSION['form_showsalepoint']){ echo '<th data-type="string"
width = 2em>Точка продажу</th>'; }
        if($_SESSION['form_showuser']){ echo '<th data-type="string" width =
3em>Користувач</th>'; }
        if($_SESSION['form_showcontract']){ echo '<th width = 2em>Договір
№</th>'; }
        if($_SESSION['form_showcomment']){ echo '<th data-type="string"
width = 3em>Коментар</th>'; }

        echo '</thead>';
        echo '<tbody>';
        while($row = mysqli_fetch_array($result)){
            echo '<tr>';
            for($i = 0; $i < mysqli_num_fields($result); $i++){
                echo "<td>".$row[$i]."</td>";
            }
            echo '</tr>';
        }
        echo '</tbody>';
    }
    else{
        echo '<script>alert("Бланків з заданими параметрами не
знайдено");window.close()</script>';
    }
    $_SESSION['form_number1'] = null;
    $_SESSION['form_number2'] = null;
    $_SESSION['form_company'] = null;
    $_SESSION['form_status'] = null;
    $_SESSION['form_user'] = null;
    $_SESSION['form_subdivision'] = null;
    $_SESSION['form_showdate'] = null;
    $_SESSION['form_showseries'] = null;
    $_SESSION['form_shownumber'] = null;
    $_SESSION['form_showproduct'] = null;
    $_SESSION['form_showcompany'] = null;
    $_SESSION['form_showstatus'] = null;

```

```

        $_SESSION['form_showsubdivision'] = null;
        $_SESSION['form_showsalepoint'] = null;
        $_SESSION['form_showuser'] = null;
        $_SESSION['form_showcontract'] = null;
        $_SESSION['form_showcomment'] = null;
    }
}
$(document).ready(function()
{
    $("#output").tablesorter();
});
</script>
</body></html>

```

## Текст підпрограми для звіту по бланкам в CSV

```

<?php
error_reporting(0);
session_start();
if($_SESSION['connected'] == false){
    header('Location: /prestige/Login.php'); // перенаправление на нужную страницу
    exit();
}
else{
    $userid = $_SESSION['userid'];
    $link = mysqli_connect($_SESSION['dbip'], $_SESSION['username'],
$_SESSION['userpassword'], $_SESSION['dbname']);
    mysqli_set_charset($link, "utf8");
    $SD = mysqli_fetch_array(mysqli_query($link, "SELECT `subdivisionid` from `my_users`
where `id` = ".$userid))[0];
    header("Content-type: text/csv");
    header("Content-Disposition: attachment; filename=Blanks ".date("Y-m-d h:m:s").".csv");
    header("Pragma: no-cache");
    header("Expires: 0");
    mb_internal_encoding("UTF-8");
    date_default_timezone_set('Europe/Kiev');
    $form_date1 = new DateTime($_POST['form_date1']);
    $form_date2 = new DateTime($_POST['form_date2']);
    $form_date2->add(new DateInterval('P1D'));
    if($_POST["form_series"]){
        $form_series = $_POST["form_series"];
        $form_series = mb_strtoupper( $form_series, 'UTF-8' );
    }
    $form_number1 = $_POST["form_number1"];
    $form_number2 = $_POST["form_number2"];
    $form_company = $_POST["form_company"];
    $form_status = $_POST["form_status"];
    $_SESSION['form_user'] = $_POST["form_user"];
    $form_salepoint = $_POST["form_salepoint"];
    $form_showdate = $_POST["form_showdate"];
    $form_showseries = $_POST["form_showseries"];
    $form_shownumber = $_POST["form_shownumber"];
    $form_showproduct = $_POST["form_showproduct"];
    $form_showcompany = $_POST["form_showcompany"];
    $form_showstatus = $_POST["form_showstatus"];
    $form_showsubdivision = $_POST["form_showsubdivision"];
    $form_showsalepoint = $_POST["form_showsalepoint"];
    $form_showuser = $_POST["form_showuser"];
    $form_showcontract = $_POST["form_showcontract"];
    $form_showcomment = $_POST["form_showcomment"];
    // Формирование запроса, выбор что выводить
    $query = "SELECT ";
    $firstfilter = true;
    if($form_showdate){ if($firstfilter) {$firstfilter = false; } else { $query =
$query.','; } $query = $query.'(select DATE_FORMAT(BSO_DateOfCreation,"%d/%m/%y"), (select
DATE_FORMAT(bso_logs.`DateChanged`, "%d/%m/%y") from bso_logs where bso_logs.`bso_id` = `bso`.`BSO_ID`
order by `DateChanged` DESC limit 1)'; }
    if($form_showseries){ if($firstfilter) {$firstfilter = false; } else { $query =
$query.','; } $query = $query.`bso`.`bso_series`; }
    if($form_shownumber){ if($firstfilter) {$firstfilter = false; } else { $query =
$query.','; } $query = $query.`bso`.`bso_number`; }

```

```

        if($form_showproduct){ if($firstfilter) {$firstfilter = false; } else { $query =
$query.','; } $query = $query.'(select ref_bso_productid.SimpleDescription from ref_bso_productid where
`id` = `bso`.`BSO_ProductID`)';}
        if($form_showcompany){ if($firstfilter) {$firstfilter = false; } else { $query =
$query.','; } $query = $query.'(select ref_bso_companyid.companyname from ref_bso_companyid where
companyid = `bso`.`BSO_CompanyID`)';}
        if($form_showstatus){ if($firstfilter) {$firstfilter = false; } else { $query =
$query.','; } $query = $query.'(select ref_bso_statusid.StatusDescription from ref_bso_statusid where
statusid = `bso`.`BSO_StatusID`)';}
        if($form_showsubdivision){ if($firstfilter) {$firstfilter = false; } else { $query =
$query.','; } $query = $query.'(SELECT ref_bso_subdivisionid.SubdivisionDescription FROM
ref_bso_subdivisionid where ref_bso_subdivisionid.SubdivisionID = (select my_users.SubdivisionID from
my_users where id = `bso`.`BSO_UserID`))';}
        if($form_showsalepoint){ if($firstfilter) {$firstfilter = false; } else { $query =
$query.','; } $query = $query.'(SELECT ref_bso_salepointid.SalePointDescription FROM ref_bso_salepointid
where ref_bso_salepointid.id = (select my_users.SalePointID from my_users where id =
`bso`.`BSO_UserID`))';}
        if($form_showuser){ if($firstfilter) {$firstfilter = false; } else { $query =
$query.','; } $query = $query.'(select my_users.names from my_users where id = `bso`.`BSO_UserID`)';}
        if($form_showcontract){ if($firstfilter) {$firstfilter = false; } else { $query =
$query.','; } $query = $query.'(REPLACE((select `dogovir`.`contract` from `dogovir` where
`dogovir`.`BlankID` = `bso`.`bso_id`), " ", ""))';}
        if($form_showcomment){ if($firstfilter) {$firstfilter = false; } else { $query =
$query.','; } $query = $query.'(select bso_logs.`Comment` from bso_logs where bso_logs.`bso_id` =
`bso`.`BSO_ID` order by `DateChanged` DESC limit 1)';}
        $query = $query.' from bso ';

        $firstwhere = true;

        if(isset($_POST['form_date1']) && $_POST['form_date1'] != ""){
            if($firstwhere){
                $query = $query." WHERE ";
                $firstwhere = false;
            }
            else{
                $query = $query." AND ";
            }
            $query = $query.' (bso.BSO_DateOfCreation >= "'. $form_date1-
>format('Y-m-j').'")';
        }
        if(isset($_POST['form_date2']) && $_POST['form_date2'] != ""){
            if($firstwhere){
                $query = $query." WHERE ";
                $firstwhere = false;
            }
            else{
                $query = $query." AND ";
            }
            $query = $query.' (bso.BSO_DateOfCreation <= "'. $form_date2-
>format('Y-m-j').'")';
        }
        if(isset($_POST['form_allseries']) == false){
            if($form_series){
                if($firstwhere){
                    $query = $query." WHERE ";
                    $firstwhere = false;
                }
                else{
                    $query = $query." AND ";
                }
                $query = $query.' (bso.bso_series = "'. $form_series.'")';
            }
            else{
                if($firstwhere){
                    $query = $query." WHERE ";
                    $firstwhere = false;
                }
                else{
                    $query = $query." AND ";
                }
                $query = $query.' (bso.bso_series = "")';
            }
        }
        if(isset($form_number1) && $form_number1 != ""){
            if($firstwhere){
                $query = $query." WHERE ";
                $firstwhere = false;
            }

```

```

    }
    else{
        $query = $query." AND ";
    }
    $query = $query.' (bso.bso_number >= "'. $form_number1.'")';
}
if(isset($form_number2) && $form_number2 != ""){
    if($firstwhere){
        $query = $query." WHERE ";
        $firstwhere = false;
    }
    else{
        $query = $query." AND ";
    }
    $query = $query.' (bso.bso_number <= "'. $form_number2.'")';
}
if(isset($_SESSION['form_company']) && $_SESSION['form_company'] != 0){
    if($firstwhere){
        $query = $query." WHERE ";
        $firstwhere = false;
    }
    else{
        $query = $query." AND ";
    }
    $query = $query.' (bso.BSO_CompanyID =
' .($_SESSION['form_company']).')';
}
if(isset($_SESSION['form_user']) && $_SESSION['form_user'] != 0){
    if($firstwhere){
        $query = $query." WHERE ";
        $firstwhere = false;
    }
    else{
        $query = $query." AND ";
    }
    $query = $query.' (bso.BSO_UserID = ' .($_SESSION['form_user']).')';
}
if($form_status != 0){
    if($firstwhere){
        $query = $query." WHERE ";
        $firstwhere = false;
    }
    else{
        $query = $query." AND ";
    }
    $query = $query.' (bso.BSO_StatusID = ' .($form_status).')';
}

if(isset($_SESSION['form_salepoint']) && $_SESSION['form_salepoint'] != 0){
    if($firstwhere){
        $query = $query." WHERE ";
        $firstwhere = false;
    }
    else{
        $query = $query." AND ";
    }
    $query = $query.' ((select my_users.SalePointID from my_users where
id = bso.BSO_UserID) = ' .($_SESSION['form_salepoint']).')';
}

// Вывод результатов в таблицу
echo (chr(0xEF) . chr(0xBB) . chr(0xBF));
$result = mysqli_query($link, $query) or die(mysqli_error($link));
$fieldCount = mysqli_num_fields($result);
$rowCount = mysqli_num_rows($result);
$numberColumnCount = 0;

$c = 0;
while ($property = mysqli_fetch_field($result)) {
    if($property->name == 'bso_number') $numberColumnCount = $c;
    if($property->name == 'bso_series') $numberSeriesCount = $c;
    $c++;
}
if($form_showdate){ echo 'Дата створення;'. 'Дата останньої зміни;'; }
if($form_showseries){ echo 'Серія;'; }
if($form_shownumber){ echo 'Номер;'; }
if($form_showproduct){ echo 'Продукт;'; }
if($form_showcompany){ echo 'Компанія;'; }

```

```

if($form_showstatus){ echo 'Стан;'; }
if($form_showsubdivision){ echo 'Підрозділ;'; }
if($form_showsalepoint){ echo 'Точка продажу;'; }
if($form_showuser){ echo 'Користувач;'; }
if($form_showcontract){ echo 'Договір №;'; }
if($form_showcomment){ echo 'Коментар;'; }
echo("\n");

for($i = 0; $i < $rowCount; $i++){
    $row = mysqli_fetch_array($result);
    for($j = 0; $j < $fieldCount; $j++){
        if($j == $numberColumnCount || $j == $numberSeriesCount) echo '=';
        echo $row[$j];
        if($j == $numberColumnCount || $j == $numberSeriesCount) echo '=';
        if($j != $fieldCount - 1) echo ' ';
    }
    if($i != $rowCount - 1) echo "\n";
}

?>

```

## Текст підпрограми для знаходження даних за ІПН

```

<?php
session_start();
$userid = $_SESSION['userid'];
$mysqli = new mysqli($_SESSION['dbip'], $_SESSION['username'],
$_SESSION['userpassword'], $_SESSION['dbname']);
if($mysqli->connect_error) {
    exit('Could not connect');
}
$mysqli->set_charset("utf8");
$sql = "SELECT count(*), `FullName`, `PhoneNumber`, `BDay`, `Adress`, `Documentid`, `Seria`, `Number`,
`DateGivement`, `WhoGave`, `ActualTo`, `DateZapysu`, `ZapysNumber`, `DocumentNumber`, `ResidenceID`, `E-
Mail`, `ClientTypeID`, `BankInfo` FROM ref_dogovir_person_persons WHERE `tin` != '999999999' and `tin` =
?";
$stmt = $mysqli->prepare($sql);
$stmt->bind_param("s", $_GET['q']);
$stmt->execute();
$stmt->store_result();
$stmt->bind_result($count, $n, $p, $b, $a, $did, $ser, $num, $DG, $WG, $AT, $DZ, $ZN, $DN, $r, $e, $c,
$bank);
$stmt->fetch();
$stmt->close();
$xml = new SimpleXMLElement('<xml/>');
$persson = $xml->addChild('person');
$persson->addChild('count', $count);
$persson->addChild('FullName', $n);
$persson->addChild('PhoneNumber', $p);
$persson->addChild('BDay', $b);
$persson->addChild('Adress', $a);
$persson->addChild('DocumentId', $did);
$persson->addChild('Ser', $ser);
$persson->addChild('Num', $num);
$persson->addChild('DG', $DG);
$persson->addChild('WG', $WG);
$persson->addChild('AT', $AT);
$persson->addChild('DZ', $DZ);
$persson->addChild('ZN', $ZN);
$persson->addChild('DN', $DN);
$persson->addChild('ResidenceID', $r);
$persson->addChild('E-Mail', $e);
$persson->addChild('ClientType', $c);
$persson->addChild('BankInfo', $bank);
Header('Content-type: text/xml');
echo($xml->asXML());

```

## ДОДАТОК В

Реалізація програмного рішення інформаційної системи  
з документообігу в страховій компанії

Опис програми

УКР.НТУУ «КПІ ім. Ігоря Сікорського»\_ТЕФ\_АПЕПС\_ТМ61

Аркушів 9

Київ – 2020

## АНОТАЦІЯ

Додаток дає можливість страховикам контролювати та оптимізувати документообіг в страховій компанії. Створений програмний продукт дає можливість:

- додавати бланки суворої звітності;
- передавати бланки суворої звітності між користувачами та автоматично формувати акт прийому-передачі;
- переглядати історію по кожному окремому бланку;
- робити звіти по бланкам використовуючи фільтри;
- створювати договори страхування на основі бланків;
- друкувати договір на бланку;
- створювати рахунок-фактуру;
- здійснювати пошук договорів за страхувальником;

Дана система розроблювалась за допомогою скриптових мов Java Script та PHP з використанням реляційної бази даних MariaDB. Програмний код писався в текстовому редакторі nano.



## ЗМІСТ

1. Загальні відомості .....	3
2. Функціональне призначення.....	4
3. Опис логічної структури.....	5
4. Використовувані технічні засоби .....	6
5. Виклик і завантаження.....	7
6. Вхідні і вихідні дані .....	8

## **ЗАГАЛЬНІ ВІДОМОСТІ**

Через використання веб інтерфейсу програмна реалізація інформаційної системи з документообігу в страховій компанії складається з декількох модулів, частина яких описана в додатку Б.

Для роботи з системою потрібно мати стабільне інтернет з'єднання, сертифікат, підписаний корінним сертифікатом безпеки, та VPN клієнт з підтримкою авторизації за ключами P12 і файлом перевірки цілісності HMAC. Для використання системи потрібно мати встановлений браузер. Особливих вимог до технічних характеристик пристрою користувача немає.

Програма написана на скриптових мовах Java Script та PHP.

## **ФУНКЦІОНАЛЬНЕ ПРИЗНАЧЕННЯ**

Розроблене програмне забезпечення створене як помічник для страхових компаній. Завдяки ньому можна чітко розподілити обов'язки людей та налагодити алгоритми і бізнес процеси в підприємстві. За допомогою цієї системи легко контролювати бланки суворої звітності, проводити облік та інвентаризацію. Монотонна робота, така як виписування рахунку-фактури, також виконується за допомогою програмного продукту. Співробітники компанії тепер не виписують договори страхування вручну, а друкують їх за допомогою системи. В разі допущення помилок – їх легко виправити в базі даних не псуючи бланк.

## ОПИС ЛОГІЧНОЇ СТРУКТУРИ

Для розмежування прав доступу до системи було розроблено механізм авторизації та аутентифікації. При завантаженні сторінок в першу чергу перевіряється чи був користувач авторизований в системі, після чого завантажується основний функціонал сторінки.

Програма складається з двох тісно пов'язаних між собою частин: одна відповідає за бланки суворої звітності, інша – договори страхування.

При написанні програмного продукту було використано бібліотеки jQuery, TableSorter, TCPDF, які знаходяться у вільному доступі.

## ВИКОРИСТОВУВАНІ ТЕХНІЧНІ ЗАСОБИ

На стороні серверу необхідно мати встановлений VPN сервер з функцією підключення з використанням сертифікатів, HTTP сервер та MySQL подібна база даних.

При створенні програмного продукту використовувався Amazon Web Service, який надає обчислювальні потужності у вигляді машин EC2 та сервіс для збереження даних S3.

Користувач в свою чергу повинен мати стабільний інтернет зв'язок, браузер, сертифікати для підключення до VPN мережі та безпосередньо сам VPN клієнт.

## **ВИКЛИК І ЗАВАНТАЖЕННЯ**

Дане програмне забезпечення не потребує завантаження, а її викликом є перехід в браузері на певну адресу.

## **ВХІДНІ І ВИХІДНІ ДАНІ**

В розробленому програмному забезпеченні вхідними даними є інформація, введена в поля форм.

В якості вихідних даних виступають:

- Акт прийому-передачі бланків суворої звітності;
- Звіти по бланкам;
- Рахунок-фактура за договором страхування;
- Договір страхування;
- Звіт по договорам страхування.